



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Análisis de redes IoT de gran escala con múltiples capas de procesamiento: una aproximación de límite fluido

TESIS PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE LA REPÚBLICA POR

Gonzalo Belcredi

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN INGENIERÍA ELÉCTRICA.

DIRECTORES DE TESIS

Dr. Ing. Pablo Belzarena..... Universidad de la República
Dr. Ing. Pablo Monzón..... Universidad de la República

TRIBUNAL

Dr. Ing. Marcelo Fiori..... Universidad de la República
Dr. Ing. Federico La Rocca..... Universidad de la República
Dr. Ing. Leonardo Steinfeld..... Universidad de la República

DIRECTOR ACADÉMICO

Dr. Ing. Pablo Monzón..... Universidad de la República

Montevideo
jueves 2 diciembre, 2021

Análisis de redes IoT de gran escala con múltiples capas de procesamiento: una aproximación de límite fluido, Gonzalo Belcredi.

ISSN 1688-2806

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).

Contiene un total de 106 páginas.

Compilada el lunes 18 julio, 2022.

<http://iie.fing.edu.uy/>

La imaginación a menudo nos llevará a mundos que nunca existieron, pero sin ella no iremos a ninguna parte.

CARL SAGAN

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

Quiero agradecer a mis tutores Pablo Belzarena y Pablo Monzón por el apoyo y la dedicación recibida estos últimos años. A pesar de trabajar gran parte del tiempo a distancia, mantuvieron mucha calidez en el trato y el trabajo con ellos fue realmente ameno.

Me gustaría también realizar una mención especial a Laura Aspirot que fue mi referencia en el método de la aproximación fluida y a quien le agradezco la invitación al Seminario de Probabilidad que me permitió intercambiar ideas sobre este trabajo.

A la Agencia Nacional de Investigación e Innovación por la financiación de la maestría sin lo cual nada de esto hubiera sido posible.

A mis colegas del Instituto de Ingeniería Eléctrica y en particular a los compañeros de la oficina “El Establo” que aunque ruidosos motivaron el trabajo en el día a día.

Y finalmente quiero agradecer a mi familia y en particular a mis padres Luis y Mónica que me tuvieron que soportar en los meses más duros de la pandemia.

Esta página ha sido intencionalmente dejada en blanco.

A Sofía y Agustín.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

En las últimas décadas con el incremento del tráfico de Internet y la aparición de nuevas aplicaciones intensivas en procesamiento surge el paradigma de Edge Computing, o computación en el borde. La idea general es derivar procesamiento hacia los bordes de la red de forma de disminuir la latencia en las comunicaciones y de esta manera disminuir también el tiempo total de ejecución de las tareas.

En el caso de redes IoT que requieran computar tareas intensivas en procesamiento, y que dispongan de infraestructura en el Edge y el Cloud se abre la interrogante de cómo distribuir ese procesamiento en las tres capas: local, Edge y Cloud.

Aún en escenarios simplificados, la decisión acerca de qué tareas procesar en cada una de las capas no es un problema sencillo o rápido de resolver. El desafío es mayor cuando se tratan de redes IoT de gran escala, con cientos o miles de dispositivos conectados.

En esta tesis se modela una red IoT de gran escala que realiza el offloading de tareas al Edge/Cloud, a través de un proceso de Markov en tiempo continuo. Con el objetivo de analizar el comportamiento del sistema, se considera una aproximación de límite fluido del proceso de Markov escalado con la cantidad de dispositivos IoT. Haciendo tender a infinito la cantidad de dispositivos, el proceso de Markov converge en probabilidad a la solución de un sistema determinístico de ecuaciones diferenciales ordinarias.

Para verificar la aproximación fluida se diseñó e implementó un simulador de eventos con la dinámica del modelo. Se pudo verificar, con un número finito de dispositivos que es posible tener una muy buena representación de la respuesta transitoria y en régimen del sistema, así como de distintas métricas de calidad de servicio según los parámetros del sistema.

A los efectos de hallar el factor óptimo de offloading se formula un problema de optimización convexa que incluye restricciones de calidad de servicio: tasa de pérdidas y tiempo de retardo.

En una segunda parte de la tesis se analiza una red IoT de gran escala con cosecha de energía (energy-harvesting). Mediante una aproximación de límite fluido se analiza la dinámica del sistema para evaluar cómo afecta la escasez de energía en la calidad de servicio, en función de los factores de offloading y la tasa de cosecha de energía.

Las simulaciones realizadas del proceso estocástico permiten validar la aproximación fluida como método para el análisis de redes IoT de gran escala en una arquitectura de computación de tres capas.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

| | |
|--|------------|
| Agradecimientos | III |
| Resumen | VII |
| 1. Introducción | 1 |
| 1.1. Contexto y motivación | 1 |
| 1.1.1. Content Delivery Networks: el comienzo de la era Edge. . . | 1 |
| 1.1.2. Aproximación fluida | 5 |
| 1.2. Principales contribuciones de la tesis | 6 |
| 1.3. Estructura del documento | 6 |
| 2. Fundamento teórico | 7 |
| 2.1. Proceso estocástico | 7 |
| 2.2. Límite fluido | 9 |
| 2.3. Sistemas autónomos | 12 |
| 2.3.1. Puntos de equilibrio y estabilidad | 13 |
| 2.4. Sistemas positivos | 14 |
| 2.5. Problemas de optimización | 15 |
| 2.5.1. Optimización convexa y dualidad | 17 |
| 3. Red IoT con múltiples capas de procesamiento | 19 |
| 3.1. Introducción | 19 |
| 3.2. Modelo del sistema | 20 |
| 3.2.1. Acceso al medio compartido | 20 |
| 3.2.2. Cantidad de nodos procesando y transmitiendo como procesos de población | 23 |
| 3.2.3. Aproximación de límite fluido | 25 |
| 3.3. Análisis del ODE determinístico | 26 |
| 3.3.1. Particiones y sistemas | 26 |
| 3.3.2. Sistema conmutado | 27 |
| 3.3.3. Análisis del Sistema I ($\mathbf{x} \in \mathcal{X}_{lin}$) | 29 |
| 3.3.4. Análisis del Sistema II ($\mathbf{x} \in \mathcal{X}_{sat}$) | 31 |
| 3.3.5. Análisis de estabilidad del sistema conmutado | 34 |
| 3.4. Calidad de Servicio | 36 |
| 3.4.1. Tasa de pérdida de paquetes | 36 |
| 3.4.2. Tiempo de retardo | 37 |

Tabla de contenidos

| | |
|--|-----------|
| 3.5. Política óptima de offloading | 39 |
| 3.6. Resultados | 41 |
| 3.6.1. Aproximación de límite fluido | 41 |
| 3.6.2. Factor óptimo de offloading | 43 |
| 3.7. Conclusiones y Trabajo Futuro | 46 |
| 4. Red IoT con Energy Harvesting | 49 |
| 4.1. Modelo del sistema | 49 |
| 4.1.1. Aproximación fluida | 53 |
| 4.2. Modelo determinístico | 54 |
| 4.2.1. ODE considerando suma constante en la variable de estados | 54 |
| 4.3. Simulación del sistema | 55 |
| 4.4. Puntos de equilibrio | 56 |
| 4.5. Estabilidad | 57 |
| 4.6. Calidad de servicio | 60 |
| 4.6.1. Salidas del sistema | 61 |
| 4.6.2. Pérdidas por energía | 61 |
| 4.6.3. Pérdidas por ocupación | 62 |
| 4.7. Casos de ejemplo | 62 |
| 4.7.1. Escenario I: descarga de red IoT | 63 |
| 4.7.2. Escenario II: carga de red IoT | 65 |
| 4.8. Conclusiones | 68 |
| Conclusiones | 73 |
| A. Proposiciones | 77 |
| B. Escenarios del sistema conmutado | 79 |
| C. Simulador del proceso estocástico | 83 |
| Referencias | 85 |
| Glosario | 87 |
| Índice de tablas | 88 |
| Índice de figuras | 90 |

Capítulo 1

Introducción

El reciente paradigma de Edge Computing (EC), o computación en el borde, en conjunto con el ya maduro Cloud Computing, se apresta a realizar una transformación radical en la calidad de servicio de aplicaciones que requieren baja latencia y procesamiento intensivo, como pueden ser: Realidad Virtual, Inteligencia Artificial, Internet of Vehicles (IoV), telemedicina, etc.

En este capítulo veremos de qué se trata Edge Computing, cuáles son sus motivaciones y los desafíos que tiene planteados actualmente. Haremos foco en particular al problema de descarga computacional (conocido en inglés como *computation offloading*) de dispositivos en el borde que llamaremos Edge Device (ED), por ejemplo dispositivos móviles o de Internet of Things (IoT), que pueden procesar sus tareas en servidores EC.

1.1. Contexto y motivación

1.1.1. Content Delivery Networks: el comienzo de la era Edge.

En los últimos años el tráfico de Internet se ha disparado con motivo de la incorporación de nuevos usuarios. De hecho, se calcula que para el 2023 se sumarán 1400 millones de usuarios respecto al año 2018, un incremento de 36 % en apenas cinco años. Asistimos además a una creciente mediación de actividades humanas a través de Internet. Ejemplos de esto son el comercio electrónico, video on-demand (streaming), juegos online, redes sociales, etc. A esto se le suma la incorporación de millones de dispositivos IoT y la comunicación Machine to Machine (M2M), que representarán el 50 % de los dispositivos de red conectados para el año 2023 según CISCO ¹.

Sin embargo, esta demanda se satisface con la misma red diseñada hace varias décadas. El cambio tecnológico en el núcleo de Internet ha demostrado ser excesivamente lento respecto al cambio que se da del lado del uso y las aplicaciones de Internet. Un ejemplo de ello es la migración entre IPv4 e IPv6 que siendo un

¹Cisco Annual Internet Report (2018–2023).

Capítulo 1. Introducción

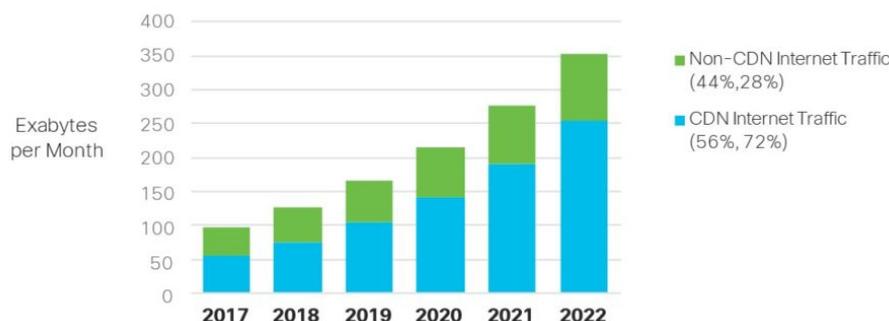


Figura 1.1: Tráfico global de Internet, años 2017 a 2022 [2].

cambio vital dado el agotamiento de las direcciones IP, fue propuesto en 1996 y la puesta en marcha demoró más de quince años.

Asistimos entonces a numerosos desafíos producidos por la explosión en el volumen y la diversidad del tráfico sobre una red que evoluciona de forma lenta: congestión de la red punto a punto, ineficiencia de los protocolos de ruteo entre sistemas autónomos como Border Gateway Protocol (BGP), redes poco fiables, ineficiencia de protocolos de comunicación como el Transmission Control Protocol (TCP) y falta de escalabilidad [1].

Uno de los primeros intentos de atender estos desafíos sin pretender un cambio en el núcleo de Internet es el concepto de Content Delivery Network (CDN) acuñado por la empresa Akamai. A los efectos de mejorar la experiencia de usuario en la navegación web, a fines de los 90' se desplegaron servidores cercanos físicamente al usuario final (denominados Edge Server (ES)) que descargaban el contenido de los sitios estáticos para evitar una eventual congestión en el *middle mile* (las conexiones intermedias que pueden atravesar distintos sistemas autónomos). Los Edge Servers envían al usuario final el caché que disponen del servidor original o bien lo obtienen y entregan a través de un sistema de transporte confiable y eficiente.

Los resultados de introducir servidores en la frontera de la red cerca del usuario final han sido tan relevantes que según CISCO se espera que en el año 2022 el 72 % del tráfico global de Internet se realice a través de CDN's (ver Figura 1.1).

Cloud, Edge y Fog Computing

Según los autores Mell y Grance en [3], el cloud computing o computación en la nube es “*un modelo que permite acceder mediante una red bajo demanda - de forma conveniente y ubicua - a un conjunto compartido de recursos informáticos configurables (por ejemplo redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar y lanzar rápidamente con mínimo esfuerzo de gestión o interacción del proveedor de servicios*”.

Si bien el cloud computing presenta distintas ventajas para sus clientes, en la reducción de costos fijos de operación y mantenimiento de infraestructura, el hecho que sea de tipo centralizado hace que por un lado tenga que recibir grandes volúmenes de información sin procesar y por otro lado es posible que el centro

1.1. Contexto y motivación

de datos se encuentre lejos físicamente de la aplicación final, con los riesgos de aumentar la latencia en la comunicación.

En este contexto surge el denominado Edge Computing, con la idea de introducir unidades de procesamiento al borde de la red cerca del dispositivo final a los efectos de disminuir la latencia en las comunicaciones.

Tres tecnologías emergen bajo el paradigma de Edge Computing: cloudlets, Mobile Edge Computing (MEC)² y Fog Computing.

Los cloudlets son cloud Data Centers pequeños ubicados en el borde de la red. Su propósito general es brindar procesamiento intensivo para aplicaciones móviles. La tecnología es promovida por Vodafone, Intel, Huawei y la Universidad Carnegie Mellon.

MEC es una tecnología que se encuentra en el proceso de estandarización por ETSI y es presentada como la evolución de las radio bases y la convergencia de tecnologías de la información con las redes de telecomunicaciones. Según el White Paper del Instituto Europeo de Normas de Telecomunicaciones (ETSI) [4], es junto con la Network Function Virtualization y el Software-Defined Networking una tecnología emergente clave para las redes 5G. Entre los casos de uso propuestos se encuentran la realidad aumentada, la aceleración inteligente de videos, IoV y el IoT. Es promovida entre otros por Huawei, IBM, Intel, Nokia y Vodafone.

La última tecnología a mencionar es la denominada Fog Computing, promovida por ARM, Cisco, Dell, Intel, Microsoft y la Universidad de Princeton nucleados en el OpenFog Consortium. El foco de aplicación es el IoT y, a diferencia de los cloudlets y MEC, procura una integración entre la infraestructura Edge y Cloud, distribuyendo los recursos en el “continuo Edge-Cloud”.

Computation offloading

Los dispositivos móviles pueden tener restricciones de procesamiento y energía por lo que requieren externalizar la computación. La migración de tareas intensivas en procesamiento desde dispositivos móviles hacia infraestructura cercana y más poderosa se denomina *computation offloading* y consta de tres aspectos clave [5]:

- *Modo de offloading y Partición de tareas.* El offloading de tareas puede ser binario o parcial. En el primer caso se envía el conjunto de tareas a procesar al ES o se ejecuta localmente. En el offloading parcial, la tarea se particiona para ejecutarse en distintos componentes. En este último caso, la partición de la tarea debe realizarse de forma óptima antes de enviarla a los ES.
- *Decisión de offloading.* Se debe decidir en qué ES procesar la tarea, o si se realiza localmente.
- *Asignación de recursos.* En esta etapa se deben distribuir los recursos de computación, comunicación y energía entre las distintas tareas o componentes.

²Rebautizado recientemente como Multi-access Edge Computing.

Capítulo 1. Introducción

Modelado de computation offloading

A la hora de modelar un problema de offloading, se deben considerar distintos escenarios y objetivos [5]:

- *Modo de offloading.* Debe especificarse si el offloading es binario o parcial.
- *Modelo de canal.* Se debe definir el modelo de canal en un medio compartido, fundamentalmente si los EDs (Edge Devices) interfieren entre sí.
- *Modelo de computación.* Aquí se analiza el costo computacional de las tareas y el modelo de colas, si existieran, cómo se modelan los tiempos de ejecución y transmisión de las tareas, así como el consumo de energía.
- *Modelo de recolección de energía.* Modelo de arribo de unidades de energía a la batería de los EDs, sean modelos determinísticos o estocásticos.

Los objetivos del offloading pueden ser:

- *Latencia mínima.* Se trata de minimizar el tiempo total de procesamiento de las tareas (ejecución y transmisión en caso de offloading).
- *Consumo energético.* En escenarios con escasez de energía se busca minimizar el consumo energético del sistema en su conjunto.
- *Pérdida de tareas.* Aquí se busca minimizar las pérdidas de tareas en el sentido de no poder ejecutar las mismas, por ejemplo, por falta de energía.
- *Tasa de procesamiento.* Dadas las restricciones de procesamiento y energía en los EDs se busca maximizar la tasa de procesamiento de tareas.
- *Eficiencia energética.* Se busca maximizar la relación de bits de tareas procesados sobre la energía consumida, i.e. [bits/Joule].
- *Minimización de costos monetarios.* En el escenario donde procesar tareas en ECs o en el cloud tenga un costo monetario se busca minimizar los costos de operación.

Los modelos de los distintos componentes del problema de offloading, en conjunto con el objetivo buscado, dan lugar a la definición de problemas de optimización para hallar la mejor política de offloading. La forma de definirlos y resolverlos es muy variada, entre las formulaciones más destacadas encontramos: optimización convexa/no-convexa, Procesos de Decisión Markoviano (MDP), Teoría de Juegos, Optimización de Lyapunov y Aprendizaje por Refuerzo [5].

Uno de los principales desafíos a la hora de resolver estos problemas es la posibilidad de escalar con la cantidad de dispositivos (EDs y ESs) así como el tiempo de resolución. Cuando un sistema de decisión trabaje en línea para resolver cómo realizar el procesamiento de las tareas, el tiempo de offloading no solo incluirá los tiempos de ejecución y transmisión sino también el tiempo de decisión del

1.1. Contexto y motivación

algoritmo de offloading, por lo que es vital reducir estos tiempos, sobre todo cuando lo que se busca con *Edge Computing* es disminuir la latencia.

En esta tesis se busca modelar una red IoT con múltiples capas de procesamiento (local-Fog-Cloud) como el límite fluido de un proceso de Markov. El objetivo final será analizar estrategias óptimas de offloading considerando restricciones de calidad de servicio.

1.1.2. Aproximación fluida

En estos problemas se suma una dificultad adicional dado que en principio no es posible conocer a priori la demanda de procesamiento o la generación de energía en los esquemas de energy-harvesting. Se trabaja entonces con variables aleatorias que modelan, por ejemplo, el tiempo entre arribos de tareas o de unidades de energía con alguna distribución de probabilidad.

De esta manera, para resolver el problema de offloading de tareas es natural pensar en modelos estocásticos para su abordaje por sobre modelos determinísticos, a los efectos de tener una representación más fiel de la realidad. Sin embargo esto es generalmente a costa de introducir complejidades en los métodos resolución.

La complejidad aumenta más aún si nuestro sistema tiene un gran número de componentes, cada uno de los cuales evoluciona según un conjunto de variables aleatorias. Si conociéramos las distribuciones de probabilidad podríamos simular el sistema, pero esto puede resultar costoso computacionalmente e incluso fuera de los tiempos requeridos por nuestra aplicación.

El campo de la física ha enfrentado problemas similares, donde una cantidad muy importante de partículas interaccionan entre sí y con ciertos componentes aleatorios. Para analizar estos sistemas complejos la física estadística, por ejemplo a través de la teoría de campo medio, propone reducir los grados de libertad del problema y considerar la existencia de un efecto medio. La idea central es que aumentando la cantidad de partículas del sistema las desviaciones alrededor de ese efecto medio se desvanecen.

Esta idea es particularmente útil en los procesos estocásticos markovianos con el denominado límite fluido. El límite consiste en acelerar el tiempo y escalar apropiadamente el proceso estocástico con cierto parámetro que se hace tender a infinito. El límite fluido mantiene las características principales del sistema estocástico original mientras que las fluctuaciones estocásticas de segundo orden se desvanecen en el procedimiento. En algunos casos, el límite fluido es una función determinística, solución de un sistema de ecuaciones diferenciales ordinarias (ODE) [6] [7].

En la medida que sea posible una formulación del problema de computation offloading como un proceso markoviano, escalable con la cantidad de dispositivos, entonces podremos estudiar a partir del límite fluido la evolución del sistema con las herramientas del análisis de ecuaciones diferenciales. Contar con una gran cantidad de EDs no será ya una dificultad como veíamos anteriormente con los modelos tradicionales, ya que el límite fluido será aún una mejor aproximación del proceso estocástico.

Capítulo 1. Introducción

1.2. Principales contribuciones de la tesis

Las principales contribuciones de esta tesis son:

- El modelado de una gran población de dispositivos IoT que se encuentran procesando, transmitiendo o en espera, como un sistema dinámico conmutado que surge de la aproximación fluida de un proceso de Markov.
- La caracterización del mencionado sistema conmutado: trayectorias, puntos de equilibrio y estabilidad en función de los parámetros del sistema.
- La formulación de un problema de optimización matemática y su resolución para calcular el factor óptimo de offloading considerando métricas de calidad de servicio como restricciones.
- El modelado de una red IoT con *energy harvesting* como un proceso de Markov y su aproximación por el límite fluido. Estudio de estabilidad del sistema determinístico resultante y de métricas de calidad de servicio.

Una parte del trabajo de esta tesis se incluyó en un artículo de conferencia para la IEEE URUCON 2021 bajo el título “*Large-Scale IoT Network Offloading to Cloud and Fog Computing: A Fluid Limit Model*” (G.Belcredi, L. Aspirot, P. Monzón, P. Belzarena).

1.3. Estructura del documento

A continuación el documento se estructura de la siguiente manera: en el capítulo 2 repasamos algunos fundamentos utilizados en el marco teórico de la tesis. En particular realizamos un resumen sobre procesos estocásticos, aproximación de límite fluido, sistemas de ecuaciones diferenciales ordinarias y optimización convexa. En el capítulo 3 veremos el primer caso de estudio que se trata del análisis de una red IoT de gran escala que puede realizar procesamiento en múltiples capas (local-Fog-Cloud). Entre las complejidades de este primer caso de estudio se destaca el modelo de canal, donde según la cantidad de usuarios transmitiendo el mismo se puede saturar. A continuación, en el capítulo 4 estudiamos el caso de una red similar pero que incorpora un modelo de *energy harvesting* con un tamaño de almacenamiento variable y donde se asume que el canal de comunicaciones se encuentra en zona lineal. Finalmente se incluyen conclusiones generales y tres apéndices con información complementaria de los casos de estudio.

Capítulo 2

Fundamento teórico

En este capítulo se presenta un resumen con los fundamentos teóricos más importantes utilizados a lo largo de la tesis. El enfoque escogido es el de resaltar, dentro de cada tema, los teoremas o resultados más relevantes para el análisis de los casos de estudio que se verán más adelante, así como familiarizar al lector con la notación. Si el lector desea profundizar en algún tema específico puede consultar en las referencias.

2.1. Proceso estocástico

Consideremos un experimento aleatorio en algún espacio muestral Ω . Una variable aleatoria X es una función que asigna un número real a cada ocurrencia en Ω .

Un proceso estocástico $\mathbf{X} = \{X(t), t \in T\}$ es una colección de variables aleatorias indexadas por el conjunto T . Para cada $t \in T$, $X(t)$ es una variable aleatoria en algún espacio de probabilidad. Cuando T es un conjunto numerable como $\{0, 1, \dots, n\}$ estamos frente a un proceso estocástico en tiempo discreto. Si T es un continuo, por ejemplo $[0, \infty)$, entonces se trata de un proceso estocástico en tiempo continuo. A su vez, el proceso toma valores en un espacio de estados S , que puede ser discreto o continuo, de una o más dimensiones. Para un proceso estocástico usaremos también la notación abreviada $\mathbf{X} = (X_t)_{t \in T}$.

Proceso de Markov en tiempo continuo

Un proceso de Markov en tiempo continuo, o cadena de Markov en tiempo continuo (en adelante CTMC por sus siglas en inglés), es un proceso estocástico $(X_t)_{t \geq 0}$ que toma valores en los enteros no negativos y que además cumple la propiedad Markoviana. Esta propiedad nos indica que la distribución condicional del estado futuro en el tiempo $t+s$, dado el estado en s y todos los estados pasados, depende únicamente del estado actual del sistema y es independiente del pasado. Expresado más formalmente, para todo tiempo $s, t \geq 0$ y enteros no negativos i, j, k , con $0 \leq u < s$ se cumple que:

Capítulo 2. Fundamento teórico

$$\begin{aligned} P(X(t+s) = j | X(s) = i, X(u) = k, 0 \leq u < s) \\ = P(X(t+s) = j | X(s) = i) \end{aligned}$$

Consideremos el caso general de una CTMC que entra a cierto estado i en un tiempo $t_0 = 0$ y que además permanece en ese estado durante un tiempo s . Queremos saber ahora la probabilidad de que la CTMC permanezca en el estado i hasta el tiempo $s+t$. Por la propiedad de Markov, la probabilidad de que la CTMC permanezca en el mismo estado durante el intervalo $[s, s+t]$ es la probabilidad no condicionada de que la CTMC permanezca en el estado i durante al menos un tiempo t [9]. Sea τ una variable aleatoria que representa el tiempo durante el cual el proceso se encuentra en el estado i antes de abandonarlo, por la propiedad Markoviana tenemos que:

$$P(\tau > s+t | \tau > s) = P(\tau > t), \quad \forall s, t \geq 0$$

La variable τ no tiene memoria y necesariamente sigue una distribución exponencial.

En la CTMC se cumple adicionalmente que el proceso, al abandonar un estado i hacia otro estado j con probabilidad de transición P_{ij} , verifica que $\sum_{j \neq i} P_{ij} = 1$.

La matriz de intensidad - o generador infinitesimal - de una CTMC (X_t) es una matriz $Q = (q_{ij}; i, j \in S)$ definida por:

$$q_{ij} = \begin{cases} \lim_{s \rightarrow 0} \frac{1}{s} P(X(s) = j | X(0) = i) & \text{si } i \neq j \\ -\sum_{j \neq i} q_{ij} & \text{sino} \end{cases}$$

Proceso de Poisson

El proceso de Poisson es un caso particular de CTMC y es muy utilizado para modelar el tiempo entre sucesos en un sistema, particularmente cuando confluyen a la entrada del sistema sucesos descoordinados entre sí. Se usa por ejemplo para modelar la ocurrencia de desastres naturales, la cantidad de automóviles que circulan por una autopista, la cantidad de clientes que ingresan a una tienda, entre otras aplicaciones. Se trata de un proceso en tiempo continuo dado que el tiempo entre sucesos toma valores en $t \geq 0, t \in \mathbb{R}$.

El proceso de Poisson es un proceso de renovación, esto es, un proceso donde la secuencia de tiempo entre sucesos es una secuencia de V.A.s IID. Una renovación es de tipo Poisson con tasa λ si el tiempo entre sucesos sigue una distribución exponencial de parámetro λ , es decir si consideramos cada elemento X_i de la sucesión de V.A.s, la densidad de probabilidad asociada es $f_{X_i}(x) = \lambda e^{-\lambda x}$.

Una forma sencilla de construir un proceso de Poisson (X_t) es considerar una secuencia S_1, S_2, \dots de V.A.s exponenciales de parámetro λ , considerar otra secuencia $J_0 = 0, J_n = S_1 + \dots + S_n$ y finalmente obtener $X(t)$ como:

$$X(t) = n \quad \text{si } J_n \leq t < J_{n+1}$$

Ejemplo 2.1.1 (Cola M/M/1 [10]). Podemos tomar como ejemplo una cola de tipo M/M/1, en la notación de Kendall significa que tanto los arribos como los servicios siguen distribuciones Poisson y que existe un solo procesador. Las tasas de arribo y servicio las fijamos como λ y μ respectivamente. El conjunto de índices T son los reales no negativos y espacio de estados $S = \{0, 1, 2, \dots\}$.

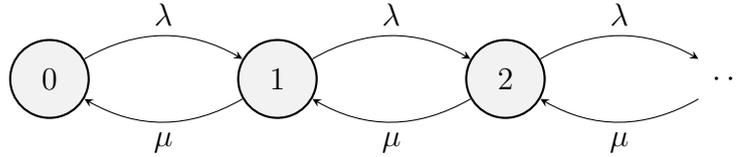


Figura 2.1: Cola M/M/1 como CTMC

Sea $X(t)$ el número de tareas en la cola o en servicio en el instante t . Entonces se puede ver que $(X_t)_{t \geq 0}$ es un proceso de Markov cuyas intensidades son $q_{i,i+1} = \lambda$, $q_{i,i-1} = \mu$ cuando $i > 0$ y $q_{i,j} = 0$ si $|i-j| > 1$, por lo que la matriz de intensidad queda como:

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & \dots & 0 \\ \mu & -(\mu + \lambda) & \lambda & \ddots & \vdots \\ 0 & \mu & -(\mu + \lambda) & \lambda & 0 \\ \vdots & \ddots & \mu & -(\mu + \lambda) & \ddots \\ 0 & \dots & 0 & \ddots & \ddots \end{pmatrix}$$

2.2. Límite fluido

El análisis de algunos procesos de Markov puede simplificarse al considerar teoremas límite, en los que tanto el tiempo como las intensidades de transición del proceso escalan convenientemente de acuerdo a cierto parámetro que se hace tender a infinito. Este parámetro según el caso puede tener cierta interpretación por ejemplo el tamaño total de una población, un área o un volumen [11].

Consideremos un proceso de Markov $(\tilde{X}^N(t))$ paramétrico en N . Su correspondiente proceso escalado $(X^N(t))$ lo definimos como:

$$(X^N(t)) = \left(\frac{\tilde{X}^N(t)}{N} \right).$$

Bajo ciertas condiciones, el proceso escalado $(X(t))$ tiende en probabilidad cuando $N \rightarrow \infty$ a un sistema determinístico mucho más sencillo de analizar. Una descripción detallada de éste método se puede encontrar en [7, 11].

Recogiendo la notación en [12], consideremos la siguiente descomposición de $\tilde{X}^N(t)$:

Capítulo 2. Fundamento teórico

$$\tilde{X}^N(t) = \tilde{X}^N(0) + \int_0^t \tilde{Q}^N(\tilde{X}^N(s))ds + \tilde{M}^N(t) \quad (2.1)$$

donde $\tilde{X}^N(0)$ es el estado inicial del proceso, $\tilde{Q}^N(l) = \sum_{m \in S} (m-l)q(l,m)$ es el *drift* del proceso, $q(l,m)$ es la intensidad desde el estado l hacia el estado m , S es el espacio de estados y $\tilde{M}^N(t)$ es una martingala¹.

Sea $X^N(t) = \tilde{X}^N(t)/N$ el proceso escalado, que a partir de (2.1) tiene la siguiente expresión:

$$X^N(t) = X^N(0) + \frac{1}{N} \int_0^t \tilde{Q}^N(\tilde{X}^N(s))ds + \frac{\tilde{M}^N(t)}{N}$$

Aplicando el Teorema 2.1 pág. 456 en [11], si existe una función Lipschitz $Q(\cdot)$ tal que:

$$\lim_{N \rightarrow \infty} \sup_{l \in S} \left\| \frac{\tilde{Q}^N(l)}{N} - Q(l/N) \right\| = 0, \quad (2.2)$$

entonces cuando $N \rightarrow \infty$ el término $\frac{\tilde{M}^N(t)}{N}$ converge a cero en probabilidad y $X^N(t)$ converge en probabilidad al proceso determinístico $x(t)$ dado por el siguiente sistema de ecuaciones diferenciales ordinarias (ODE):

$$\dot{x} = Q(x(t))$$

Ejemplo 2.2.1 (Epidemia, adaptado de [11]). *En una población se desata una epidemia y nos interesa conocer la evolución de las personas susceptibles (que pueden enfermarse) e infectadas (que pueden transmitir la enfermedad), que llamaremos N_s y N_i respectivamente. La población total de individuos es N y se asume constante.*

La transmisión de la enfermedad ocurre cuando un susceptible tiene un encuentro con una persona infectada. Aunque no siempre se produce el contagio sino que es un evento aleatorio con cierta probabilidad. La velocidad de propagación de la enfermedad dependerá de la probabilidad de que interactuemos con una persona infectada así como de la probabilidad de que en un encuentro un infectado transmita efectivamente la enfermedad a un susceptible.

Analícemos en primer lugar el problema razonando en términos de valores medios y tomándonos alguna licencia en la rigurosidad matemática. Podemos esperar que una persona se infecte a una tasa $\rho \frac{N_i}{N}$, donde ρ es un parámetro que regula el factor de replicación del virus (número de encuentros sociales, contagiosidad, etc.) y $\frac{N_i}{N}$ es la densidad de personas infectadas sobre el total de la población.

¹Una martingala (Y_t) es un proceso estocástico particular que tiene como propiedad que $\mathbb{E}(\|Y(t)\|) < +\infty$.

2.2. Límite fluido

Considerando el total de personas susceptibles podemos esperar que en un tiempo ΔT ocurran en media un total de $N_s \rho \frac{N_i}{N} \Delta T$ nuevas infecciones, es decir que $\Delta N_s = -\rho N_s \frac{N_i}{N} \Delta T$.

Veamos ahora cómo varía en media la cantidad de personas infectadas. Naturalmente que las personas que dejan de ser susceptibles es porque fueron infectadas, que es el cálculo que realizamos anteriormente. Lo que debemos considerar ahora es el decremento de infectados debido a la recuperación de las personas enfermas. Supongamos que una persona se recupera con tasa κ , entonces en cierto ΔT esperamos que en media se recuperen $\kappa N_i \Delta T$ personas enfermas, por lo cual el balance de infectados queda como $\Delta N_i = (\rho N_s \frac{N_i}{N} - \kappa N_i) \Delta T$.

Considerando las densidades $x_1 = N_s/N$ y $x_2 = N_i/N$ y tomando un ΔT lo suficientemente chico podemos aproximar la dinámica de la epidemia por el siguiente conjunto de ecuaciones diferenciales:

$$\begin{aligned}\frac{dx_1}{dt} &= -\rho x_1 x_2 \\ \frac{dx_2}{dt} &= \rho x_1 x_2 - \kappa x_2\end{aligned}$$

Este primer análisis intuitivo acerca de cómo se comportaría la epidemia analizando los efectos medios, podemos formalizarlo mediante el análisis de límite fluido presentado anteriormente.

Consideremos el proceso de Markov $(\tilde{X}_t(l))$ con estado $l = (N_s, N_i) \in \mathbb{N}^2$. Supongamos que el tiempo entre infecciones y el tiempo entre recuperaciones son variables aleatorias con distribución exponencial con las tasas presentadas previamente. La intensidad de nuevas infecciones queda entonces como:

$$\tilde{q}_{(N_s, N_i), (N_s-1, N_i+1)}^N = \rho N_s \frac{N_i}{N} = N \rho \frac{N_s}{N} \frac{N_i}{N}$$

Las personas infectadas se recuperan luego de cierto tiempo y logran inmunidad frente a la enfermedad, la intensidad de recuperación queda dada por:

$$\tilde{q}_{(N_s, N_i), (N_s, N_i-1)}^N = \kappa N_i = N \kappa \frac{N_i}{N}$$

El drift está dado por $\tilde{Q}^N(l) = \sum_{m \in S} (m - l) q(l, m)$. En este caso los estados m y l pertenecen a \mathbb{N}^2 , y como transiciones posibles tenemos que un estado puede saltar desde (N_s, N_i) hacia $(N_s - 1, N_i + 1)$ en el caso de una infección, y desde (N_s, N_i) hacia $(N_s, N_i - 1)$ en el caso de una recuperación, por lo que el drift del proceso es:

$$\tilde{Q}^N(N_s, N_i) = \begin{pmatrix} -1 \\ 1 \end{pmatrix} N \rho \frac{N_s}{N} \frac{N_i}{N} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} N \kappa \frac{N_i}{N} = \begin{pmatrix} -N \rho \frac{N_s}{N} \frac{N_i}{N} \\ N \rho \frac{N_s}{N} \frac{N_i}{N} - N \kappa \frac{N_i}{N} \end{pmatrix}$$

Consideremos ahora la siguiente función $Q : \mathbb{R}^2 \rightarrow \mathbb{R}^2$:

Capítulo 2. Fundamento teórico

$$Q(x, y) = \begin{pmatrix} -\rho xy \\ \rho xy - \kappa y \end{pmatrix}$$

$Q(\frac{N_s}{N}, \frac{N_i}{N})$ se puede escribir de la siguiente manera:

$$Q\left(\frac{N_s}{N}, \frac{N_i}{N}\right) = \begin{pmatrix} -\rho \frac{N_s}{N} \frac{N_i}{N} \\ \rho \frac{N_s}{N} \frac{N_i}{N} - \kappa \frac{N_i}{N} \end{pmatrix}$$

Por un lado la función Q es Lipschitz dado que sus derivadas parciales quedan acotadas en cualquier compacto de \mathbb{R}^2 . Por otra parte, como $\frac{\tilde{Q}^N(l)}{N} = Q(l/N)$ se cumple el límite que aparece en la Ec. (2.2). Finalmente y dado que se cumplen las hipótesis del Teorema 2.1 pág. 456 en [11], el sistema determinístico asociado a este modelo de epidemia queda como:

$$\begin{aligned} \dot{x}_1 &= -\rho x_1 x_2 \\ \dot{x}_2 &= \rho x_1 x_2 - \kappa x_2, \end{aligned}$$

Donde x_1 y x_2 son las variables escaladas $\frac{N_s}{N}$ y $\frac{N_i}{N}$ respectivamente.

En la Figura 2.2 se puede apreciar que la trayectoria dada por el ODE con condición inicial $x_0 = (1, 0)$ es una muy buena aproximación respecto a la simulación estocástica de la epidemia tomando una población de 500 habitantes.

En este ejemplo queda más clara la interpretación del drift como la tendencia media del proceso.

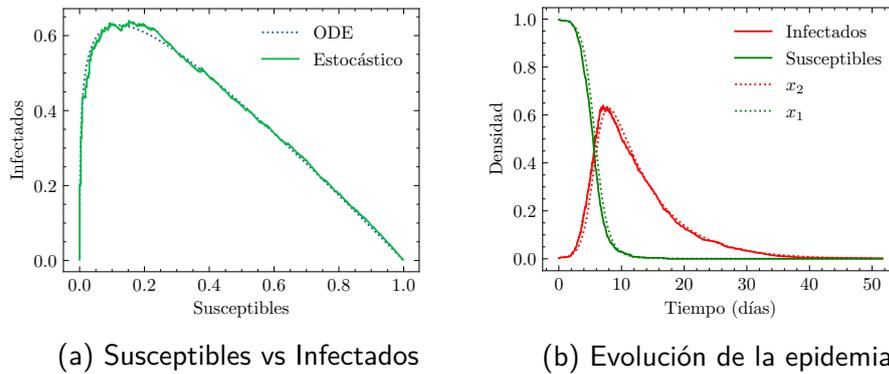


Figura 2.2: Aproximación fluida del Ejemplo 2.2.1 con parámetros $N = 500$, $\rho = 1.2$ y $\kappa = 1/7$.

2.3. Sistemas autónomos

En esta tesis se trabaja en general con sistemas de ecuaciones diferenciales lineales del tipo:

$$\begin{cases} \dot{x} = f(x), & \text{con } x(0) = x_0 \\ f(x) = Ax + u \end{cases} \quad (2.3)$$

donde $x \in \mathbb{R}^n$ es nuestro vector de estados, $A \in \mathcal{M}^{n \times n}$ es la matriz fundamental y $u \in \mathbb{R}^n$ es una entrada constante. Dado que el término de la derecha en la ecuación diferencial no depende del tiempo se trata de un sistema autónomo.

2.3.1. Puntos de equilibrio y estabilidad

El punto de equilibrio \tilde{x} se halla resolviendo $f(\tilde{x}) = \mathbf{0}$. Si la matriz fundamental es invertible, entonces existe un único punto de equilibrio que se puede calcular de forma matricial como $\tilde{x} = -A^{-1}u$. Si la matriz no es invertible entonces el sistema tiene infinitos puntos de equilibrio.

Decimos que el punto de equilibrio \tilde{x} es *estable* para una ecuación diferencial autónoma $\dot{x} = f(x)$ si para todo $\epsilon > 0$ existe $\delta > 0$ tal que si $x_0 \in \mathcal{B}(\tilde{x}, \delta)$ entonces la solución de la ecuación diferencial con condición inicial $x(0) = x_0$ cumple que $x(t) \in \mathcal{B}(\tilde{x}, \epsilon) \quad \forall t \geq 0$. De lo contrario decimos que \tilde{x} es *inestable*.

Un punto de equilibrio es *asintóticamente estable* si es estable y además $\exists \mathcal{B}(\tilde{x}, \delta_0)$ tal que todas las soluciones $x(t)$ de la ecuación con condición inicial $x(0) = x_0 \in \mathcal{B}(\tilde{x}, \delta_0)$ cumplen que:

$$\lim_{t \rightarrow \infty} x(t) = \tilde{x}$$

La estabilidad del punto de equilibrio en los sistemas autónomos se puede estudiar analizando los valores propios de la matriz fundamental. En particular, si todos los valores propios tienen parte real no positiva entonces el punto de equilibrio es estable. Si todos los valores propios tienen parte real negativa, hay un único equilibrio, y es asintóticamente estable.

El análisis de estabilidad mediante los valores propios se simplifica mucho cuando trabajamos en \mathbb{R}^2 . Para una matriz 2×2 la traza y su determinante definen los dos valores propios y, por lo tanto, la estabilidad o no de los eventuales equilibrios. En efecto, consideremos la siguiente matriz genérica:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Para calcular los valores propios debemos resolver la ecuación $\det(A - \lambda I) = 0$. Con a, b, c y d reales tenemos que $\det(A - \lambda I) = \lambda^2 - (a + d)\lambda + (ad - bc) = \lambda^2 - \text{Tr}(A)\lambda + \det(A)$. Las raíces quedan entonces de la forma

$$\lambda = \frac{1}{2} \left[\text{Tr}(A) \pm \sqrt{\text{Tr}(A)^2 - 4 \det(A)} \right]$$

Capítulo 2. Fundamento teórico

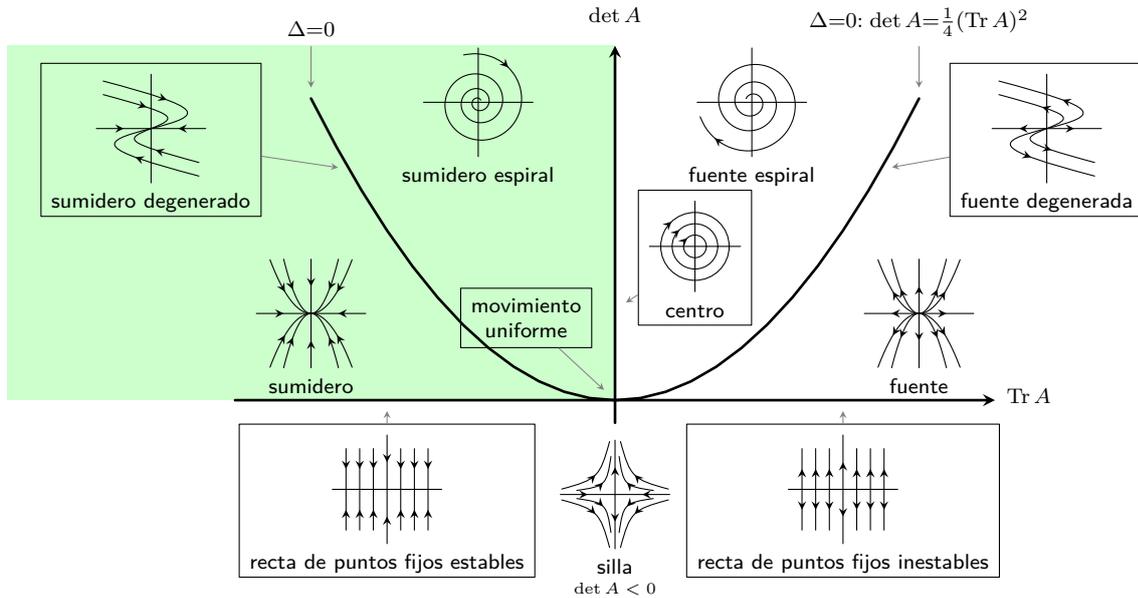


Figura 2.3: Diagrama de Poincaré. Adaptado de [15]

Las raíces son reales cuando $\text{Tr}^2 \geq 4 \det$ y complejas conjugadas si $\text{Tr}^2 < 4 \det$. Además, el producto de los valores propios equivale al determinante de la matriz, y la suma de los valores propios a la traza. Con esta propiedad se puede ver que los valores propios, sean reales o una pareja de complejos, tendrán parte real estrictamente negativa si se cumple simultáneamente que $\text{Tr}(A) < 0$ y $\det(A) > 0$.

En la Figura 2.3 se muestra el diagrama de Poincaré que clasifica los diagramas de fase en el plano $\text{Tr} - \det$. En verde se marca la región donde los puntos de equilibrio son estables (la región incluye los ejes).

2.4. Sistemas positivos

Como veremos más adelante en la sección 4.5, los resultados y teoremas de los sistemas positivos serán de vital importancia para concluir acerca de la estabilidad del sistema determinístico que surge de la aproximación fluida del proceso de Markov en el segundo caso de estudio (energía en red IoT), donde una de las dificultades radica en que el espacio de estados tiene dimensión arbitraria.

Nos interesa estudiar un caso particular de sistemas lineales, los denominados sistemas positivos. Vamos a considerar a continuación sistemas descritos por la siguiente ecuación de estados:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + bu(t) \\ y(t) &= c^T x(t)\end{aligned}$$

Los sistemas positivos son aquellos cuyas variables de estado toman solo valores no negativos. Entre ejemplos de sistemas positivos podemos encontrar modelos

2.5. Problemas de optimización

estocásticos donde las variables de estado son no negativas porque representan una probabilidad, sistemas de almacenamiento, procesos industriales involucrando reactores químicos, modelos económicos y poblacionales, etc.

A continuación incluimos una definición formal y algunas propiedades. Nos basamos en [16], donde puede encontrarse un estudio detallado de esta clase de sistemas.

Definición 2.4.1 (Sistema positivo). *Un sistema lineal (A, b, c^T) se dice positivo si y solo si para todo estado inicial no negativo y para toda entrada no negativa, el estado y la salida son siempre no negativos.*

Definición 2.4.2 (Matriz Metzler). *Una matriz $A = (a_{ij})$ es Metzler si todos sus elementos fuera de la diagonal son no negativos, es decir $a_{ij} \geq 0, \forall(i, j), i \neq j$.*

Teorema 2.4.3. *Un sistema lineal (A, b, c^T) es positivo si y solo si A es una matriz Metzler.*

Teorema 2.4.4 (Matriz reducible/irreducible, pág. 671 [17]). *Una matriz cuadrada $A_{n \times n}$ es reducible si existe una matriz de permutación P tal que:*

$$P^T A P = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}, \text{ donde } X \text{ e } Y \text{ son cuadradas.}$$

De otra manera la matriz A se dice irreducible. Además, una matriz A es irreducible si y solo si el grafo asociado $G(A)$ es fuertemente conexo.

Teorema 2.4.5 (Perron-Frobenius para matrices Metzler). *Sea A un matriz Metzler irreducible y su conjunto de valores propios $\sigma(A)$. Entonces:*

- *el número real $\mu(A) = \max(\text{Re}(\lambda), \lambda \in \sigma(A))$ es un valor propio simple de A ;*
- *Sea x_F el vector propio de Frobenius asociado al valor propio $\mu(A)$, entonces $x_F \succ 0$ y es único (más allá de multiplicaciones por escalares positivos)²;*
- *Si x es un vector propio de A con $x \succ 0$, entonces $Ax = \mu(A)x$.*

2.5. Problemas de optimización

En esta tesis estaremos interesados en resolver problemas de optimización como el siguiente:

$$\begin{aligned} \min_{x \in \mathcal{D}} \quad & f(x) \\ \text{s.a} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m. \\ & h_j(x) = 0, \quad j = 1, \dots, p. \end{aligned} \tag{2.4}$$

²Usaremos la notación $(.) \succ 0$ para referirnos a un vector con todas sus entradas positivas.

Capítulo 2. Fundamento teórico

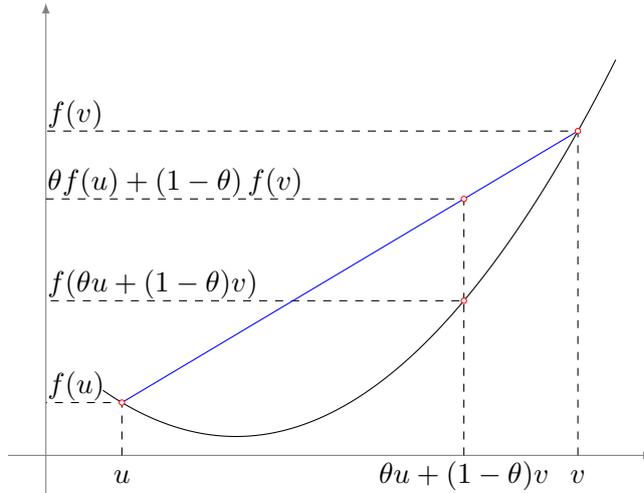


Figura 2.4: Función convexa

En esta notación, el vector $x \in \mathcal{D}$ es la variable de optimización, la función $f : \mathcal{D} \rightarrow \mathbb{R}$ es la función objetivo y las ecuaciones $g_i(x) \leq 0$ y $h_j(x) = 0$ son restricciones de desigualdad e igualdad respectivamente.

Definición 2.5.1 (Conjunto convexo). *Un conjunto \mathcal{C} es convexo si el segmento que une dos puntos cualquiera de \mathcal{C} pertenece a \mathcal{C} , es decir dados $x_1, x_2 \in \mathcal{C}$ y $\forall \theta$ con $0 \leq \theta \leq 1$, se tiene que $\theta x_1 + (1 - \theta)x_2 \in \mathcal{C}$.*

Definición 2.5.2 (Función convexa). *Una función $f : \mathcal{D} \rightarrow \mathbb{R}$ es convexa si el dominio \mathcal{D} de $f(\cdot)$ es un convexo y si para todo $u, v \in \mathcal{D}$ y para todo θ con $0 \leq \theta \leq 1$ se tiene que:*

$$f(\theta u + (1 - \theta)v) \leq \theta f(u) + (1 - \theta)f(v)$$

Esta condición puede visualizarse mejor en la Fig. 2.4:

Una propiedad importante de los problemas de optimización convexa es que cualquier óptimo local es también un óptimo global. De esta manera, métodos iterativos sencillos como el descenso por gradiente (y en caso de restricciones en su versión de gradiente proyectado) que permiten encontrar óptimos locales, en los problemas de optimización convexa permiten encontrar el óptimo global.

Cuando la función objetivo f es diferenciable y el problema presenta restricciones, entonces un punto x es óptimo si y solo si x pertenece al conjunto factible \mathcal{X} (el conjunto que resulta de intersectar el dominio de f y los conjuntos de restricciones) y además x verifica que:

$$\nabla f(x)^T (y - x) \geq 0 \quad \forall y \in \mathcal{X}$$

En este caso, si $\nabla f(x) \neq 0$ quiere decir que $-\nabla f(x)$ define un hiperplano que soporta el conjunto factible \mathcal{X} . Cuando el problema no tiene restricciones la condición de optimalidad se reduce a la más conocida $\nabla f(x) = 0$.

2.5. Problemas de optimización

Para que el problema de optimización 2.4 sea convexo se requieren las siguientes condiciones:

- La función objetivo es convexa.
- Las funciones de desigualdad $g_i(x)$ son todas funciones convexas.
- Las funciones $h_j(x)$ son afines en la variable de optimización, es decir que son de la forma $h_j(x) = a_j^T x + b_j$.

El valor óptimo x^* del problema de optimización (2.4) se define como:

$$x^* = \inf\{f(x) \mid g_i(x) \leq 0, i = 1, \dots, m, h_j(x) = 0, j = 1, \dots, p\}$$

2.5.1. Optimización convexa y dualidad

Consideremos nuevamente el problema de optimización (2.4). El Lagrangeano se define como:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \mu_j h_j(x) \quad (2.5)$$

en donde λ_i es el multiplicador de Lagrange asociado a la restricción de desigualdad $f_i(x) \leq 0$ y μ_j es el multiplicador asociado a la restricción de igualdad $g_j(x) = 0$.

Ahora estamos en condiciones de definir la función dual del Lagrangeano $g : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ como:

$$g(\lambda, \mu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \mu) \quad (2.6)$$

Se define el *problema dual* como:

$$\begin{array}{ll} \text{mín} & g(\lambda, \mu) \\ \lambda, \mu & \\ \text{s.a} & \lambda \succeq 0 \end{array} \quad (2.7)$$

Los vectores $\lambda = [\lambda_1, \dots, \lambda_m]$ y $\mu = [\mu_1, \dots, \mu_p]$ se denominan variables duales y en este contexto (2.4) es el *problema primal*. Como $\lambda_i \geq 0$ para $i = 1, \dots, m$ y dado que $g_i(x) \leq 0$ y $h_j(x) = 0$, entonces se puede ver que la función dual es una cota inferior del valor óptimo, es decir:

$$g(\lambda, \mu) \leq f(x^*) \quad \forall \lambda, \mu \quad (2.8)$$

Capítulo 2. Fundamento teórico

La diferencia $g(\lambda, \mu) - f(x)$ se denomina diferencia (o *gap*) de dualidad. Las denominadas *condiciones de Karush-Kuhn-Tucker (KKT)* establecen condiciones necesarias y suficientes de optimalidad cuando las funciones f, g_i, h_j son diferenciables. En particular, para el caso donde el problema primal (2.4) es convexo, si x^*, λ^*, μ^* verifican las siguientes condiciones:

$$f_i(x^*) \leq 0 \quad (2.9)$$

$$h_j(x^*) = 0 \quad (2.10)$$

$$\lambda_i^* \geq 0 \quad (2.11)$$

$$\lambda_i^* f_i(x^*) = 0 \quad (2.12)$$

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) = 0 \quad (2.13)$$

entonces x^* es solución del problema primal y la pareja (λ^*, μ^*) es solución del problema dual. Se cumple además que el gap de dualidad es cero ($g(\lambda^*, \mu^*) = f(x^*)$), por lo que se puede resolver el problema primal resolviendo el problema dual.

Respecto a los multiplicadores de Lagrange queremos realizar dos comentarios. En primer lugar es frecuente darle una interpretación *económica* a los multiplicadores de Lagrange, viéndolos como precios asociados a las restricciones. En segundo lugar, se puede deducir a partir de la condición (2.12), referida como *complementary slackness*, que si una restricción no se encuentra activa, es decir, que $f_i(x^*) < 0$, entonces el multiplicador λ_i^* debe anularse. Al resolver un problema de optimización con muchas restricciones, los multiplicadores son de gran utilidad ya que pueden indicarnos de forma inmediata qué restricciones se activaron en la solución del problema de optimización.

Capítulo 3

Red IoT con múltiples capas de procesamiento

3.1. Introducción

En los últimos años, la cantidad de dispositivos IoT se ha incrementado considerablemente, y esta tendencia se espera que continúe y se acelere con la aparición de las redes de quinta generación [19]. Nuevas aplicaciones y servicios requerirán un aumento en la capacidad de procesamiento y una reducción de la latencia: realidad aumentada, internet de los vehículos, dispositivos de monitoreo médico, etc. Para resolver estos desafíos, *Fog Computing* emerge como un habilitador tecnológico.

Mientras que el *Cloud Computing* dispone de capacidades aumentadas de recursos, el procesamiento local en los dispositivos IoT o en el Fog puede permitir una reducción en la latencia y en el costo de las comunicaciones. Sin embargo, estos últimos dispositivos están limitados en sus recursos de energía, memoria, CPU, etc. De esta manera, un problema relevante es determinar un factor óptimo de offloading en las tres capas: local, Fog y Cloud, y de esta manera maximizar la performance del sistema en su conjunto. La literatura en esta área presenta varios modelos estocásticos de offloading. Los artículos [20] [21] analizan el estado del arte en este tema. Los autores en [22] [23] [24] proponen Modelos de Markov para el offloading de procesamiento. Hasta donde sabemos previamente no se han presentado trabajos que analicen el comportamiento asintótico de una red IoT de gran escala que realiza offloading al Fog/Cloud en un medio de acceso compartido.

En esta tesis se propone una estrategia para determinar el factor óptimo de offloading de una red IoT de gran escala. Las principales contribuciones en este sentido son:

- El modelado de una gran población de dispositivos IoT que se encuentran procesando, transmitiendo o en espera, como un sistema dinámico conmutado que surge de la aproximación fluida de un Proceso de Markov.
- La caracterización del mencionado sistema conmutado: trayectorias, puntos de equilibrio y estabilidad en función de los parámetros del sistema.

Capítulo 3. Red IoT con múltiples capas de procesamiento

- La formulación de un problema de optimización matemática y su resolución para calcular el factor óptimo de offloading considerando métricas de calidad de servicio como restricciones.

3.2. Modelo del sistema

Un conjunto de dispositivos IoT (sensores, móviles, M2M) reciben tareas que requieren procesamiento. Este puede realizarse localmente en el dispositivo IoT o en infraestructura Fog o Cloud (ver Figura 3.1).

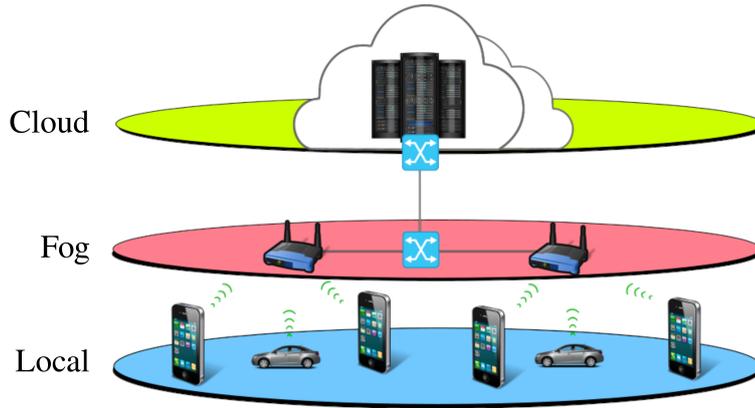


Figura 3.1: Arquitectura de offloading de tres capas (Local-Fog-Cloud).

Supongamos que llega una nueva tarea y el dispositivo IoT se encuentra disponible (ni procesando ni transmitiendo). En este caso, la tarea es procesada con probabilidad α (el factor de offloading) o transmitida de forma inalámbrica al nodo Fog con probabilidad $1 - \alpha$. Se asume que los nodos no cuentan con un *buffer* para guardar tareas, por lo que las tareas que arriban cuando el nodo está ocupado se pierden. Tanto el tiempo de procesamiento como el de transmisión son V.A.s con distribución exponencial.

Sea N la cantidad total de dispositivos IoT; $\tilde{N}_{tx}^N(t)$, $\tilde{N}_p^N(t)$ y $\tilde{N}_{idle}^N(t)$ son los procesos estocásticos que representan la cantidad de dispositivos IoT que se encuentran transmitiendo, procesando y en espera respectivamente, en tiempo t con $\tilde{N}_{tx}^N(t) + \tilde{N}_p^N(t) + \tilde{N}_{idle}^N(t) = N$.

3.2.1. Acceso al medio compartido

Introducción

Un aspecto fundamental del modelado es poder tener una estimación razonables del tiempo de transmisión, elemento que repercutirá en la calidad de servicio y en el punto de operación del sistema. El throughput que se puede alcanzar en un canal de comunicaciones, además de las características intrínsecas del canal, depende de las distintas tecnologías empleadas para la comunicación, en particular de la modulación, la codificación, etc.

3.2. Modelo del sistema

De forma de poder generalizar modelos de throughput, más allá de la tecnología empleada, es frecuente utilizar el Teorema de Shannon-Hartley [25] que determina una cota superior a la tasa de información para una comunicación sin errores dado el ancho de banda disponible y la relación señal a ruido.

Los autores en [26], a partir de la expresión de la cota máxima de capacidad del canal, formulan un modelo de throughput donde una cantidad \mathcal{L} de dispositivos interfieren entre sí. El throughput para el dispositivo l , con $1 \leq l \leq \mathcal{L}$ está dado por:

$$r_l = B \log_2 \left(1 + \frac{p_l h_l^2}{N_0 + \sum_{i \neq l} p_i h_i^2} \right), \quad (3.1)$$

siendo B el ancho de banda del canal, N_0 la densidad espectral de potencia del ruido, y para el dispositivo l , p_l es la potencia de transmisión y h_l la ganancia del canal (entre el dispositivo l y el nodo Fog que lo atiende). Los demás dispositivos transmitiendo tienen potencia p_i con ganancia de canal h_i , con $i \neq l$.

A los efectos de nuestro problema nos interesará estimar el throughput agregado de la red variando la cantidad de dispositivos IoT transmitiendo. Buscamos estudiar por tanto una función $R(\mathcal{L}) = \sum_{l=1}^{\mathcal{L}} r_l$, y analizar su comportamiento cuando la cantidad de dispositivos es muy grande.

Razonando a partir de la ecuación (3.1), supongamos que todos los dispositivos son idénticos y con el mismo canal, es decir $h_l = h$ y $p_l = p$ para $l = 1, \dots, \mathcal{L}$, se puede ver que:

$$\lim_{\mathcal{L} \rightarrow +\infty} R(\mathcal{L}) = \lim_{\mathcal{L} \rightarrow +\infty} \mathcal{L} B \log_2 \left(1 + \frac{p h^2}{N_0 + (\mathcal{L} - 1) p h^2} \right) = \frac{B}{\ln(2)}.$$

Este fenómeno se puede visualizar en la Figura 3.2 que muestra $R(\mathcal{L})$ con $0 \leq \mathcal{L} \leq 500$ y parámetros $B = 10\text{MHz}$, $N_0 = 10^{-6}\text{ W}$, $h = 10^{-2}$ y $p = 10^{-3}\text{ W}$. Se puede ver que en una primera etapa el throughput agregado crece fuertemente con la cantidad de dispositivos transmitiendo hasta estabilizarse en $R = 14.3\text{ Mbps} \simeq \frac{B}{\ln(2)}$.

El modelo basado en la ecuación (3.1) si bien logra capturar el fenómeno de interferencia entre transmisiones simultáneas no contempla el escenario donde se utilizan mecanismos de acceso al medio como los que se usan habitualmente en este tipo de redes.

Por lo general cuando un conjunto de dispositivos comparten un mismo medio físico se implementan mecanismos de acceso para evitar colisiones y mejorar la performance del enlace. Para esto lo que se hace es multiplexar en tiempo o frecuencia las comunicaciones en base a algún protocolo. Una clase de mecanismos usados habitualmente son aquellos basados en el acceso aleatorio como lo son Carrier Sense Multiple Access (CSMA) o ALOHA por la simplicidad del protocolo.

Podemos tomar como ejemplo el protocolo de acceso aleatorio CSMA en su esquema más simple, el CSMA 1-persistente. Cuando un transmisor tiene un paquete a transmitir, en primer lugar escucha el canal para verificar que no haya nadie transmitiendo en ese momento. Si el canal se encuentra disponible entonces

Capítulo 3. Red IoT con múltiples capas de procesamiento

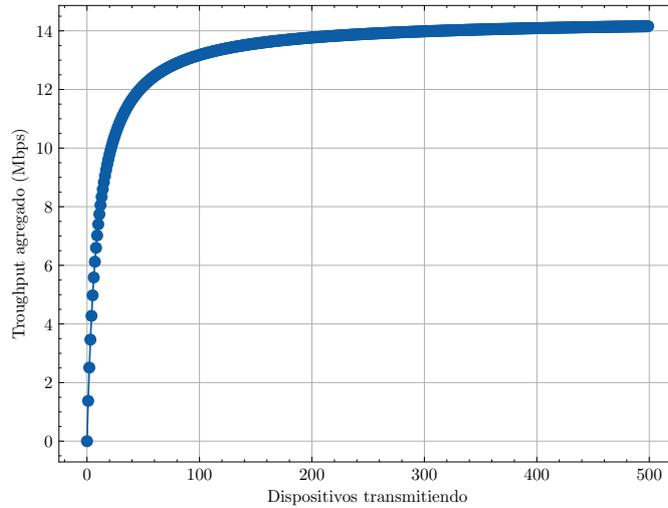


Figura 3.2: $R(\mathcal{L})$, throughput agregado en función de la cantidad de dispositivos transmitiendo

comienza a transmitir. En cambio, si el canal se encuentra ocupado espera hasta que el canal quede disponible e inmediatamente después inicia su transmisión. Si en esa transmisión ocurre una colisión entonces la estación espera un tiempo aleatorio antes de repetir la transmisión. Este es el caso en el mecanismo 1-persistente dado que la estación comienza a transmitir apenas encuentra el canal disponible (con probabilidad 1).

En CSMA p-persistente el tiempo se discretiza en slots y si la estación tiene un paquete a transmitir y el canal se encuentra disponible, la transmisión se realiza con probabilidad p en ese mismo slot y con probabilidad $1 - p$ en el slot siguiente. Un análisis en profundidad de los distintos mecanismos de acceso al medio y cuestiones relativas a la estabilidad y performance puede encontrarse en [27].

Hay dos formas de desperdiciar capacidad del canal, por un lado cuando los transmisores tienen paquetes para enviar pero el canal no está en uso (por ejemplo aguardando un temporizador de backoff antes de transmitir) y en segundo lugar cuando se producen colisiones.

La performance de mecanismos de acceso al medio puede visualizarse en un diagrama throughput vs carga, donde la carga es la tasa media de intentos de transmisión. En la Figura 3.3 se muestra la performance de distintos mecanismos de acceso al medio en algunas variantes de CSMA y ALOHA.

Cuando la carga es baja el canal puede acomodar fácilmente la demanda, pero si la carga aumenta a partir de cierto punto los tiempos de espera cobran relevancia y disminuye la performance del enlace. A continuación introduciremos un modelo de throughput que aproxima este comportamiento mediante una función lineal a tramos.

3.2. Modelo del sistema

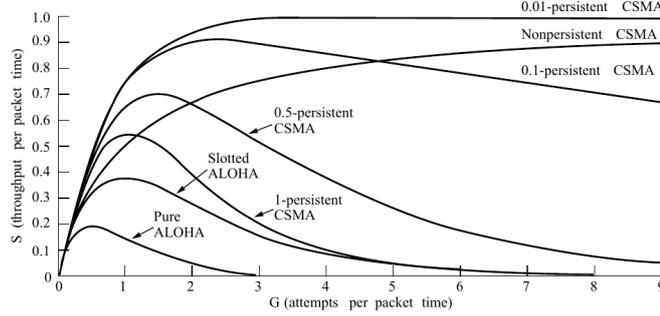


Figura 3.3: Throughput vs Carga de distintos protocolos de acceso al medio [28]

Modelo propuesto

El canal de comunicaciones es compartido con los nodos IoT y tiene una función de throughput $\tilde{C}^N(\tilde{N}_{tx}^N)$ escalable con N . La red IoT se divide en L subredes, cada una atendida por un gateway Fog de forma que $N \propto L$. En cada subred, un gateway puede atender hasta M dispositivos con throughput constante k y el máximo throughput que puede brindar el gateway a la subred es $c_g = Mk$.

En cada subred, que distinguiremos con el índice $i \in \{1, 2, \dots, L\}$, se implementa un mecanismo de control de acceso al medio (ALOHA, CSMA, etc.). A partir de las consideraciones realizadas en la introducción de esta sección 3.2.1 y suponiendo que la carga G_i es proporcional a la cantidad de usuarios transmitiendo $\tilde{N}_{i,tx}^N$ en la subred i , consideramos la aproximación de throughput que se muestra en la Figura 3.4a. La aproximación es lineal a tramos con pendientes $k > 0$ en la denominada zona lineal y pendiente $-k_1 < 0$ en la denominada zona saturada.

Consideraremos ahora el throughput agregado de la red IoT $\tilde{C}^N(\tilde{N}_{tx}^N)$ como la suma de los throughputs de cada una de las subredes, las que consideraremos idénticas entre sí. Por conveniencia definiremos el throughput máximo por dispositivo IoT como $c = c_g L / N$. El throughput agregado tiene entonces la siguiente expresión:

$$\tilde{C}^N(\tilde{N}_{tx}^N) = \begin{cases} k\tilde{N}_{tx}^N, & \text{si } 0 \leq \tilde{N}_{tx}^N \leq cN/k \\ cN(1 + \frac{k_1}{k}) - k_1\tilde{N}_{tx}^N, & \text{si } cN/k < \tilde{N}_{tx}^N \leq N \end{cases} \quad (3.2)$$

Inicialmente, el throughput se incrementa con la cantidad de usuarios transmitiendo, hasta que se alcanza un valor de throughput máximo y la performance del enlace disminuye (ver Figura 3.4b).

3.2.2. Cantidad de nodos procesando y transmitiendo como procesos de población

Podemos pensar el sistema como un proceso de población con una cantidad fija de “partículas” similares de diferentes clases. Estas partículas en este caso son

Capítulo 3. Red IoT con múltiples capas de procesamiento

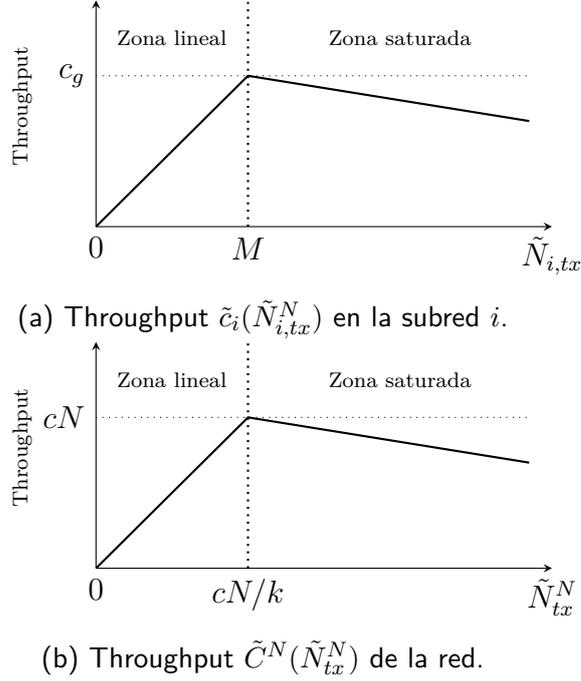


Figura 3.4: Escalado del throughput agregado de la red con N .

dispositivos IoT, cada uno clasificado según su estado: transmitiendo, procesando o en espera.

La tasa a la que cada clase evoluciona depende de las transiciones entre los diferentes estados. A modo de ejemplo, consideremos la intensidad del nacimiento de nodos procesando. El nacimiento ocurre cuando arriba una nueva tarea a cualquiera de los nodos en espera, y además el controlador de offloading decide procesar la tarea localmente. Dado que el tiempo entre arribos sigue una distribución exponencial de parámetro λ , el tiempo hasta que una primera tarea arribe a cualquiera de los nodos en espera sigue una distribución exponencial de parámetro λN_{idle} . Si consideramos el factor de offloading, entonces la intensidad de nacimientos de nodos procesando es $\alpha \lambda \tilde{N}_{idle}^N$. De forma similar, la población de nodos procesando decrece en una unidad cuando culmine el procesamiento de cualquiera de las tareas actualmente siendo procesadas. En este caso, la intensidad de muertes de nodos procesando es $\mu \tilde{N}_p^N$, donde μ es la tasa de procesamiento.

Análogamente, la intensidad de nacimientos de nodos transmitiendo es $(1 - \alpha) \lambda \tilde{N}_{idle}^N$. La tasa de transmisión de un único dispositivo (tareas transmitidas por segundo) puede calcularse como el throughput disponible por usuario $\tilde{C}^N(\tilde{N}_{tx}^N) / \tilde{N}_{tx}^N$ dividido por $1/v$ que es el tamaño del paquete a transmitir de la correspondiente tarea. De esta manera, el tiempo hasta que culmine una primera transmisión entre todas las que se realicen actualmente se distribuye de forma exponencial con parámetro $v \tilde{N}_{tx}^N \tilde{C}^N(\tilde{N}_{tx}^N) / \tilde{N}_{tx}^N = v \tilde{C}^N(\tilde{N}_{tx}^N)$.

Resumiendo, el proceso de Markov $(\tilde{N}_p^N(t), \tilde{N}_{tx}^N(t))$ tiene tasas de transición $\tilde{q}((\tilde{N}_p^N, \tilde{N}_{tx}^N), (\tilde{N}'_p, \tilde{N}'_{tx}))$ desde el estado $(\tilde{N}_p^N, \tilde{N}_{tx}^N)$ hacia el estado $(\tilde{N}'_p, \tilde{N}'_{tx})$,

| Parámetro | Descripción |
|--------------|--|
| N | Cantidad de dispositivos IoT |
| M | Cantidad de gateways Fog |
| λ | Tasa de arribo de tareas |
| μ | Tasa de procesamiento (local) |
| μ_e | Tasa de procesamiento (Fog o Cloud) |
| c_g | Throughput máximo por gateway |
| k | Throughput por usuario transmitiendo (zona lineal) |
| k_1 | Parámetro de saturación |
| v | Relación tareas transmitidas por bit |
| δ | Tasa máxima de procesamiento máxima en gateway Fog |
| β_1 | Precio de procesamiento en Fog |
| β_2 | Precio de procesamiento en Cloud |
| ϵ | Tasa Máxima de Pérdidas (QoS) |
| τ_{max} | Tiempo de Retardo Máximo (QoS) |

Tabla 3.1: Listado y definición de parámetros.

definido por:

$$\tilde{q}((\tilde{N}_p^N, \tilde{N}_{tx}^N), (\tilde{N}_p^N + 1, \tilde{N}_{tx}^N)) = \alpha\lambda(N - \tilde{N}_p^N - \tilde{N}_{tx}^N) \quad (3.3)$$

$$\tilde{q}((\tilde{N}_p^N, \tilde{N}_{tx}^N), (\tilde{N}_p^N - 1, \tilde{N}_{tx}^N)) = \mu\tilde{N}_p^N \quad (3.4)$$

$$\tilde{q}((\tilde{N}_p^N, \tilde{N}_{tx}^N), (\tilde{N}_p^N, \tilde{N}_{tx}^N + 1)) = (1 - \alpha)\lambda(N - \tilde{N}_p^N - \tilde{N}_{tx}^N) \quad (3.5)$$

$$\tilde{q}((\tilde{N}_p^N, \tilde{N}_{tx}^N), (\tilde{N}_p^N, \tilde{N}_{tx}^N - 1)) = v\tilde{C}^N(\tilde{N}_{tx}^N) \quad (3.6)$$

3.2.3. Aproximación de límite fluido

Estamos interesados en evaluar la performance del sistema con un número muy importante de dispositivos IoT como función del factor de offloading. Aunque se pueden realizar simulaciones con distintos parámetros, el comportamiento asintótico puede ser más fácilmente estudiado con un sistema determinístico equivalente.

Para este propósito introducimos una aproximación de límite fluido del proceso de Markov $(\tilde{N}_p^N(t), \tilde{N}_{tx}^N(t))$. Este procedimiento, cuya motivación fue presentada en la sección 2.2, puede leerse con más detalle en [7, 11, 12].

Como primer paso para la aproximación, definimos las variables como densidades: $n_{tx}^N = \tilde{N}_{tx}^N/N$, $n_p^N = \tilde{N}_p^N/N$ y $n_{idle}^N = \tilde{N}_{idle}^N/N$ tal que $0 \leq n_{tx}^N \leq 1$ y $0 \leq n_p^N \leq 1$. Definimos también la función de throughput por dispositivo IoT, $C(n_{tx}^N) = \frac{\tilde{C}^N(\tilde{N}_{tx}^N)}{N}$ dada por:

Capítulo 3. Red IoT con múltiples capas de procesamiento

$$C(n_{tx}^N) = \begin{cases} kn_{tx}^N, & \text{si } 0 \leq n_{tx}^N \leq c/k \\ c(1 + \frac{k_1}{k}) - k_1 n_{tx}^N, & \text{si } c/k < n_{tx}^N \leq 1 \end{cases} \quad (3.7)$$

En la proposición 3.2.1 aplicamos el teorema presentado en 2.2 al proceso de Markov $(\tilde{N}_p^N(t), \tilde{N}_{tx}^N(t))$.

Proposición 3.2.1. *El proceso escalado $(n_p^N(t), n_{tx}^N(t)) = \frac{1}{N}(\tilde{N}_p^N(t), \tilde{N}_{tx}^N(t))$ converge en probabilidad cuando $N \rightarrow \infty$ a $(n_p(t), n_{tx}(t))$, solución del siguiente ODE determinístico:*

$$(\dot{n}_p, \dot{n}_{tx}) = Q(n_p, n_{tx}) \quad (3.8)$$

donde $Q(\cdot)$ está dado por:

$$Q(n_p, n_{tx}) = \begin{pmatrix} \alpha\lambda(1 - n_p - n_{tx}) - \mu n_p \\ (1 - \alpha)\lambda(1 - n_p - n_{tx}) - vC(n_{tx}) \end{pmatrix}$$

Demostración. El drift del proceso con intensidades \tilde{q} definidas en Eqs. (3.3)-(3.6) se calcula como $\tilde{Q}^N(l) = \sum_{m \in S} (m - l)\tilde{q}(l, m)$. Para $l = (l_p, l_{tx}) \in S$:

$$\frac{\tilde{Q}^N(l_p, l_{tx})}{N} = \begin{bmatrix} \alpha\lambda(1 - l_p/N - l_{tx}/N) - \mu l_p/N \\ (1 - \alpha)\lambda(1 - l_p/N - l_{tx}/N) - vC(l_{tx}/N) \end{bmatrix}$$

Dado que $\frac{\tilde{Q}^N(l_p, l_{tx})}{N} = Q(l_p/N, l_{tx}/N)$ se verifica lo siguiente:

$$\lim_{N \rightarrow \infty} \sup_{l \in S} \left\| \frac{\tilde{Q}^N(l_p, l_{tx})}{N} - Q(l_p/N, l_{tx}/N) \right\| = 0$$

Además se tiene que $Q(\cdot)$ es una función Lipschitz, entonces por el Teorema 2.1 pág. 456 en [11] el proceso escalado $(n_p^N(t), n_{tx}^N(t))$ tiende en probabilidad al ODE dado por $(\dot{n}_p, \dot{n}_{tx}) = Q(n_p, n_{tx})$. □

3.3. Análisis del ODE determinístico

3.3.1. Particiones y sistemas

A partir de la proposición 3.2.1, la dinámica del sistema determinístico queda dada por:

$$\begin{cases} \dot{n}_p & = \alpha\lambda(1 - n_{tx} - n_p) - n_p\mu \\ \dot{n}_{tx} & = (1 - \alpha)\lambda(1 - n_{tx} - n_p) - vC(n_{tx}) \\ n_p(t_0) & = n_p^0 \\ n_{tx}(t_0) & = n_{tx}^0 \end{cases} \quad (3.9)$$

3.3. Análisis del ODE determinístico

La función de throughput (3.7) hace que el ODE definido en la ecuación (3.9) sea un sistema lineal a tramos. Se consideran entonces dos sistemas lineales según el valor de n_{tx} . Consideramos la siguiente partición en poliedros que se muestra en la Figura 3.5:

- $\mathcal{X}_{lin} = \{(n_p, n_{tx}) | 0 \leq n_{tx} \leq c/k, 0 \leq n_p \leq 1, n_p + n_{tx} \leq 1\}$
- $\mathcal{X}_{sat} = \{(n_p, n_{tx}) | c/k \leq n_{tx} \leq 1, 0 \leq n_p \leq 1, n_p + n_{tx} \leq 1\}$.

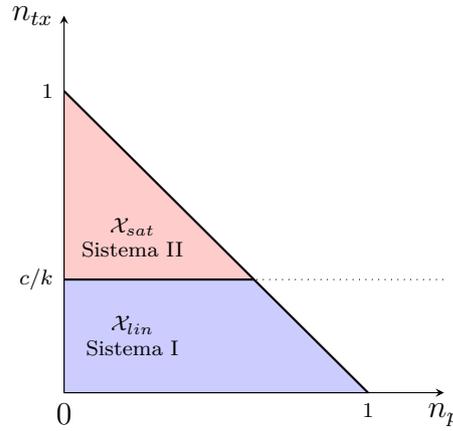


Figura 3.5: Regiones del sistema conmutado

Cuando $\mathbf{x} \in \mathcal{X}_{lin}$ el canal se encuentra en zona lineal y el throughput vale kn_{tx} , en este caso la red tiene la dinámica que denominaremos **Sistema I**. En cambio, si $\mathbf{x} \in \mathcal{X}_{sat}$ el canal está en zona de saturación y la red tiene la dinámica del **Sistema II** y el throughput vale $c(1 + \frac{k_1}{k}) - k_1 n_{tx}$.

3.3.2. Sistema conmutado

Se modela la dinámica general como un sistema conmutado [29] que sigue una secuencia $s = [(i_0, t_0), (i_1, t_1), \dots, (i_k, t_k), \dots, (i_N, t_N)]$ entre dos sistemas lineales de primer orden del tipo:

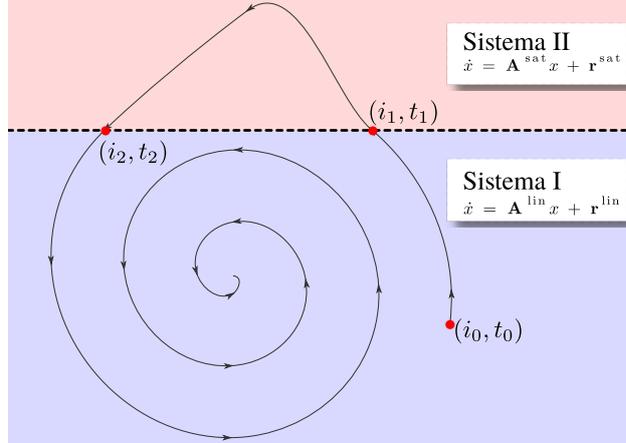
$$\dot{\mathbf{x}}(t - t_k) = \mathbf{A}_{i_k} \mathbf{x}(t - t_k) + \mathbf{r}_{i_k}, \quad (3.10)$$

para $t_k \leq t < t_{k+1}$ donde $\mathbf{x} = (n_p, n_{tx})^T$, \mathbf{A}_{i_k} es la matriz fundamental, \mathbf{r}_{i_k} es una entrada constante. En la Figura 3.6 se muestra un ejemplo del sistema conmutado, la trayectoria que se muestra es simplemente con fines ilustrativos.

Cuando $\mathbf{x} \in \mathcal{X}_{lin}$ entonces $i_k = 1$ y el **Sistema I** se encuentra activo. Si $\mathbf{x} \in \mathcal{X}_{sat}$ entonces $i_k = 2$ y el **Sistema II** se encuentra activo. En lo que sigue notaremos $\mathbf{A}^{lin} = \mathbf{A}_1$, $\mathbf{A}^{sat} = \mathbf{A}_2$, $\mathbf{r}^{lin} = \mathbf{r}_1$ y $\mathbf{r}^{sat} = \mathbf{r}_2$ y las matrices valen:

Capítulo 3. Red IoT con múltiples capas de procesamiento

Figura 3.6: Una trayectoria del sistema conmutado queda definida por la secuencia de índices y las dinámicas de los sistemas. En este ejemplo, la secuencia de conmutación es $[(1, t_0), (2, t_1), (1, t_2)]$. El sistema conmutado comienza con la dinámica del Sistema I y conmuta en $t = t_1$ al Sistema II. Luego en $t = t_2$ la trayectoria ingresa a la región inferior por lo que el sistema conmuta por última vez al Sistema I. Finalmente la trayectoria converge hacia un punto de equilibrio estable en la región inferior.



$$\mathbf{A}^{\text{lin}} = \begin{bmatrix} -\alpha\lambda - \mu & -\alpha\lambda \\ -(1-\alpha)\lambda & -kv - (1-\alpha)\lambda \end{bmatrix}, \quad \mathbf{r}^{\text{lin}} = \begin{bmatrix} \alpha\lambda \\ (1-\alpha)\lambda \end{bmatrix}$$

$$\mathbf{A}^{\text{sat}} = \begin{bmatrix} -\alpha\lambda - \mu & -\alpha\lambda \\ -(1-\alpha)\lambda & k_1v - (1-\alpha)\lambda \end{bmatrix}, \quad \mathbf{r}^{\text{sat}} = \begin{bmatrix} \alpha\lambda \\ (1-\alpha)\lambda - c(1+k_1/k)v \end{bmatrix}$$

El objetivo final será encontrar el factor de offloading que maximice el valor esperado de cierta función de utilidad. Para ello vamos a encontrar el punto de equilibrio del sistema y analizar su estabilidad. Si el equilibrio es asintóticamente estable, entonces tendremos un funcionamiento en régimen del sistema con un equilibrio de dispositivos en espera, transmitiendo y procesando.

Con este propósito, en las secciones 3.3.3 y 3.3.4 analizamos cada subsistema por separado **sin restricciones de dominio** para luego estudiar el efecto de la conmutación.

Proposición 3.3.1. *Si la función $C(n_{tx})$ es no negativa, el triángulo factible \mathcal{X} es invariante¹ frente a la dinámica (3.8).*

Demostración. Consideremos en primer lugar una condición inicial de la forma $[0, n_{tx}^0]^T$, con $0 \leq n_{tx}^0 \leq 1$. El campo tiene su primera coordenada positiva:

$$[\alpha\lambda(1 - n_{tx}^0) \quad (1 - \alpha)\lambda(1 - n_{tx}^0) - C(n_{tx}^0)v]^T$$

Consideremos ahora una condición inicial de la forma $[n_p^0, 0]^T$, con $0 \leq n_p^0 \leq 1$. El campo tiene su segunda coordenada positiva:

$$[\alpha\lambda(1 - n_p^0) - kn_p^0 \quad (1 - \alpha)\lambda(1 - n_p^0)]^T$$

¹ Un conjunto $\mathcal{C} \subseteq \mathcal{R}^n$ es invariante frente a la dinámica del sistema autónomo $\dot{x} = f(x)$ si para cada trayectoria $x, x(t) \in \mathcal{C} \implies x(\tau) \in \mathcal{C} \quad \forall \tau \geq t$.

3.3. Análisis del ODE determinístico

Finalmente, consideremos una condición inicial de la forma $[n_p^0, n_{tx}^0]^T$, con $n_p^0 + n_{tx}^0 = 1$. El campo está dado por:

$$[-\mu n_p^0 \quad -C(n_{tx}^0)v]^T$$

La primer coordenada es negativa mientras que la segunda es no positiva y la trayectoria ingresará al interior del triángulo. \square

Dado que el triángulo factible es invariante frente a la dinámica, la trayectoria nunca lo abandonará para cada condición inicial factible. En este sentido, el sistema conmutado refleja lo que sucede en el mundo físico: la suma de dispositivos procesando, transmitiendo y en espera nunca puede superar la cantidad total de dispositivos IoT.

3.3.3. Análisis del **Sistema I** ($\mathbf{x} \in \mathcal{X}_{lin}$)

La dinámica en esta zona está dada por:

$$\dot{\mathbf{x}} = \mathbf{A}^{lin}\mathbf{x} + \mathbf{r}^{lin}, \quad (3.11)$$

con matrices:

$$\mathbf{A}^{lin} = \begin{bmatrix} -\alpha\lambda - \mu & -\alpha\lambda \\ -(1-\alpha)\lambda & -kv - (1-\alpha)\lambda \end{bmatrix}, \quad \mathbf{r}^{lin} = \begin{bmatrix} \alpha\lambda \\ (1-\alpha)\lambda \end{bmatrix}$$

El sistema tiene un único punto de equilibrio que verifica $\dot{\mathbf{x}}(\hat{n}_p^{lin}, \hat{n}_{tx}^{lin}) = 0$ dado que $\det(\mathbf{A}^{lin}) \neq 0$ como se ve en la proposición 3.3.3. El punto de equilibrio tiene por coordenadas:

$$\hat{n}_p^{lin} = \frac{\alpha}{\frac{\mu}{\lambda} + \frac{\mu}{kv} + \alpha(1 - \frac{\mu}{kv})}$$

$$\hat{n}_{tx}^{lin} = \frac{1 - \alpha}{1 + \frac{kv}{\lambda} + \alpha(\frac{kv}{\mu} - 1)}$$

Lugar geométrico de los puntos de equilibrio variando el factor de offloading

La función $\Gamma^{lin}(\alpha) = (\hat{n}_p^{lin}(\alpha), \hat{n}_{tx}^{lin}(\alpha))$ mapea el intervalo $\alpha \in [0, 1]$ en un segmento de recta \mathbf{C}^{lin} incluido en el triángulo factible (ver Figura 3.7).

Proposición 3.3.2. *Sea \mathbf{C}^{lin} el conjunto de los puntos de equilibrio del **Sistema I** definido en (3.11) con $\alpha \in [0, 1]$. \mathbf{C}^{lin} es un segmento de recta definido por:*

$$\mathbf{C}^{lin} = \{(\hat{n}_p, \hat{n}_{tx}) | \hat{n}_{tx} = \frac{\lambda}{\lambda+kv} - \frac{\lambda+\mu}{\lambda+kv}\hat{n}_p, 0 \leq \hat{n}_p \leq \frac{\lambda}{\lambda+\mu}\}$$

Capítulo 3. Red IoT con múltiples capas de procesamiento

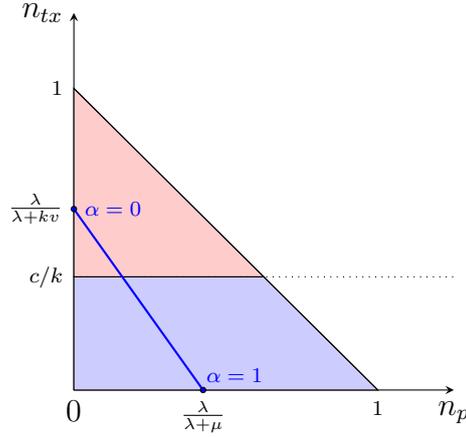


Figura 3.7: Lugar geométrico del punto de equilibrio del Sistema I variando α . \mathbf{C}^{lin} se muestra en azul y corresponde a un segmento de recta incluido en el triángulo factible.

Demostración. Haciendo que $\dot{n}_p, \dot{n}_{tx} = 0$ en (3.11) obtenemos:

$$\tilde{n}_{tx} = \frac{1 - \alpha}{\alpha} \frac{\mu}{kv} \tilde{n}_p \quad (3.12)$$

$$\tilde{n}_p = \frac{\alpha \lambda}{\mu + \alpha \lambda (1 - \mu/kv) + \lambda \mu/kv} \quad (3.13)$$

Los puntos de equilibrio se pueden representar en el plano complejo como $\Gamma(\alpha) = \tilde{n}_p(\alpha) + j\tilde{n}_{tx}(\alpha)$ y al substituir esta expresión por (3.12) y (3.13) obtenemos:

$$\Gamma(\alpha) = \frac{\alpha \lambda (1 - j\mu/kv) + j\mu \lambda/kv}{\alpha \lambda (1 - \mu/kv) + \mu + \lambda \mu/kv}$$

Vemos que los puntos de equilibrio pueden expresarse como el mapeo de $\alpha \in \mathcal{R}$ por la transformación de Möbius $\Gamma(z) = \frac{az+b}{cz+d}$, con $a = \lambda(1 - j\mu/kv) \in \mathbb{C}$, $b = j\mu \lambda/kv \in \mathbb{C}$, $c = \lambda(1 - \mu/kv) \in \mathbb{R}$ y $d = \mu + \lambda \mu/kv \in \mathbb{R}$. Dado que $\text{Im}(c) = 0$ y $\text{Im}(d) = 0$, a partir de la proposición A.0.1 tenemos que los puntos de equilibrio quedan incluidos en una recta. También se puede ver que $-d/c \notin [0, 1]$ y por lo tanto $\Gamma(\alpha) \neq \infty \quad \forall \alpha \in [0, 1]$.

Podemos ahora calcular $m, e \in \mathbb{R}$ tal que $\hat{n}_{tx} = m\hat{n}_p + e$. La pendiente se calcula como $m = \frac{\text{Im}[\Gamma(z=1)] - \text{Im}[\Gamma(z=0)]}{\text{Re}[\Gamma(z=1)] - \text{Re}[\Gamma(z=0)]} = -\frac{\lambda + \mu}{\lambda + kv}$ y el offset $e = \text{Im}[\Gamma(z=0)] = \frac{\lambda}{\lambda + kv}$. Además, y dado que la transformación de Möbius es continua, el mapeo del segmento de recta $[0, 1] \in \mathcal{R}$ es otro segmento de recta sobre el plano complejo (ver Figura 3.7):

$$\mathbf{C}^{\text{lin}} = \{(\hat{n}_p, \hat{n}_{tx}) \mid \hat{n}_{tx} = \frac{\lambda}{\lambda + kv} - \frac{\lambda + \mu}{\lambda + kv} \hat{n}_p, 0 \leq \hat{n}_p \leq \frac{\lambda}{\lambda + \mu}\}$$

□

Estabilidad del punto de equilibrio

Proposición 3.3.3. *El Sistema I definido en (3.11) sin restricciones de dominio es asintóticamente estable.*

Demostración. Dado que la matriz fundamental $\mathbf{A}^{\text{lin}} \in \mathcal{M}_{2 \times 2}$, como vimos en la sección 2.3.1, los valores propios γ_1 y γ_2 tienen parte real negativa exactamente cuando $\text{tr}(\mathbf{A}^{\text{lin}}) < 0$ y $\det(\mathbf{A}^{\text{lin}}) > 0$. En efecto,

$$\begin{aligned}\text{tr}(\mathbf{A}^{\text{lin}}) &= -\lambda - \mu - kv < 0 \quad \text{y} \\ \det(\mathbf{A}^{\text{lin}}) &= \alpha\lambda kv + k\mu v + \mu(1 - \alpha)\lambda > 0,\end{aligned}$$

por lo que el sistema es asintóticamente estable. \square

 3.3.4. Análisis del Sistema II ($\mathbf{x} \in \mathcal{X}_{\text{sat}}$)

La dinámica en esta zona está dada por:

$$\dot{\mathbf{x}} = \mathbf{A}^{\text{sat}}\mathbf{x} + \mathbf{r}^{\text{sat}}, \quad (3.14)$$

con matrices:

$$\mathbf{A}^{\text{sat}} = \begin{bmatrix} -\alpha\lambda - \mu & -\alpha\lambda \\ -(1 - \alpha)\lambda & k_1v - (1 - \alpha)\lambda \end{bmatrix}, \quad \mathbf{r}^{\text{sat}} = \begin{bmatrix} \alpha\lambda \\ (1 - \alpha)\lambda - c(1 + k_1/k)v \end{bmatrix}$$

En los casos donde $\alpha \neq \frac{\mu(\lambda - k_1v)}{\lambda(\mu + k_1v)}$ la matriz \mathbf{A}^{sat} es invertible y existe un único punto de equilibrio que verifica $\dot{\mathbf{x}}(\hat{n}_p^{\text{sat}}, \hat{n}_{tx}^{\text{sat}}) = 0$ y tiene por coordenadas:

$$\begin{aligned}\hat{n}_p^{\text{sat}} &= \frac{\alpha\lambda}{\mu + \alpha\lambda} \frac{k_1v - c(1 + k_1/k)v}{k_1v + (1 - \alpha)\lambda \left(\frac{\alpha\lambda}{\mu + \alpha\lambda} - 1 \right)} \\ \hat{n}_{tx}^{\text{sat}} &= \frac{c(1 + k_1/k)v + (1 - \alpha)\lambda \left(\frac{\alpha\lambda}{\mu + \alpha\lambda} - 1 \right)}{k_1v + (1 - \alpha)\lambda \left(\frac{\alpha\lambda}{\mu + \alpha\lambda} - 1 \right)}\end{aligned}$$

Proposición 3.3.4. *El Sistema II sin restricciones de dominio es asintóticamente estable si y solo si se cumple que:*

$$(I) \quad k_1v < \lambda + \mu$$

$$(II) \quad \alpha < \frac{\mu(\lambda - k_1v)}{\lambda(\mu + k_1v)}$$

Demostración. Dado que la matriz fundamental $\mathbf{A}^{\text{sat}} \in \mathcal{M}_{2 \times 2}$, los valores propios tendrán parte real estrictamente negativa cuando $\text{tr}(\mathbf{A}^{\text{sat}}) < 0$ y $\det(\mathbf{A}^{\text{sat}}) > 0$.

$$\begin{aligned}\text{tr}(\mathbf{A}^{\text{sat}}) &= -\lambda - \mu + k_1v < 0 \iff k_1v < \lambda + \mu \\ \det(\mathbf{A}^{\text{sat}}) &= (1 - \alpha)\mu\lambda - k_1v(\alpha\lambda + \mu) > 0 \iff \alpha < \frac{\mu(\lambda - k_1v)}{\lambda(\mu + k_1v)}\end{aligned}$$

\square

Capítulo 3. Red IoT con múltiples capas de procesamiento

Lugar geométrico de los puntos de equilibrio y estabilidad variando el factor de offloading

Por conveniencia vamos a considerar la función $\Gamma : [0, 1] \subset \mathbb{R} \rightarrow \mathbb{C}$ que mapea el factor de offloading al punto de equilibrio del **Sistema II** en el plano complejo:

$$\Gamma(\alpha) = \tilde{n}_p(\alpha) + j\tilde{n}_{tx}(\alpha)$$

En la proposición 3.3.5 mostramos que $\Gamma : z \in \mathbb{C} \rightarrow \mathbb{C}$ es una transformación de Möbius, es decir que se puede escribir de la forma $\Gamma(z) = \frac{az+b}{cz+d}$. Vemos además que el conjunto $\mathbf{C}^{\text{sat}} = \{\Gamma(\alpha) | \alpha \in [0, 1]\}$ compuesto por los puntos de equilibrio del **Sistema II** queda incluido en una recta en el plano complejo.

Como veremos a continuación, el signo de $k_1v - \lambda$ será clave para determinar las características de \mathbf{C}^{sat} y la estabilidad de los puntos de equilibrio. Para el análisis nos será útil la ubicación de los puntos de equilibrio en $\alpha = 0$ y $\alpha = 1$. Definimos por lo tanto los puntos $L_0 = (\tilde{n}_p|_{\alpha=0}, \tilde{n}_{tx}|_{\alpha=0})$ y $L_1 = (\tilde{n}_p|_{\alpha=1}, \tilde{n}_{tx}|_{\alpha=1})$ que tienen como coordenadas:

$$L_0 = \left(0, \frac{(1 + k_1/k)cv - \lambda}{k_1v - \lambda} \right) \quad (3.15)$$

$$L_1 = \left(\frac{\lambda}{\mu + \lambda} \frac{k_1v - (1 + k_1/k)cv}{k_1v}, \left(\frac{1}{k_1} + \frac{1}{k} \right) c \right) \quad (3.16)$$

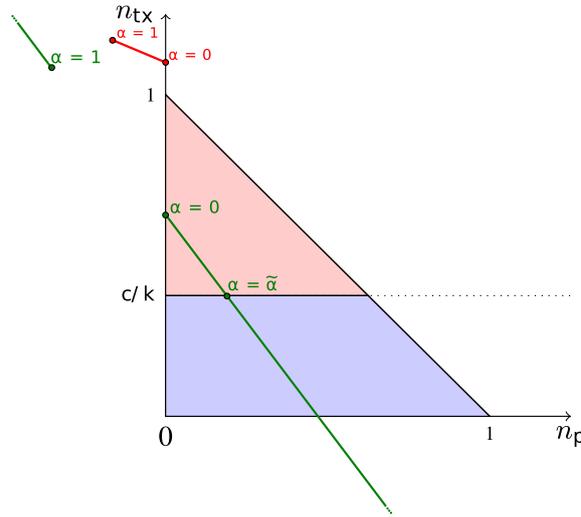


Figura 3.8: \mathbf{C}^{sat} en dos escenarios distintos. El segmento de recta en rojo corresponde al lugar geométrico del punto de equilibrio con $\alpha \in [0, 1]$ cuando $k_1v > \lambda$, y en verde el lugar geométrico cuando $k_1v \leq \lambda$.

Caso cuando $k_1v \leq \lambda$:

A partir de la proposición 3.3.5, considerando que $\Gamma(-\frac{d}{c}) = \infty$ en una transformación de Möbius y que en nuestro caso $-\frac{d}{c} \in \mathbb{R}$, se desprende que en algunos casos la función $\Gamma(\cdot)$ tiene un polo $\alpha_p = \frac{\mu(\lambda - k_1v)}{\lambda(k_1v + \mu)}$ en el intervalo $[0, 1] \in \mathbb{R}$.

3.3. Análisis del ODE determinístico

En efecto, si $k_1v \leq \lambda$ entonces $\alpha_p \in [0, 1]$ (ver proposición A.0.2). El conjunto \mathcal{C}^{sat} en este caso será disconexo como el que se muestra en verde en la Figura 3.8. El punto L_0 se ubica en el eje n_{tx} , y cuando se incrementa α el punto de equilibrio se mueve con pendiente $\frac{\lambda+\mu}{k_1v-\lambda}$ mientras que $\alpha < \alpha_p$, y luego cuando $\alpha > \alpha_p$ vuelve por la parte superior del plano hasta llegar al punto L_1 . El caso donde $\alpha = \alpha_p$ corresponde a la condición de no invertibilidad de la matriz fundamental (ver comienzo de la sección 3.3.4).

Además se pueden realizar las siguientes apreciaciones:

- Si $\alpha < \alpha_p$ el Sistema II es estable dado que se cumplen las dos condiciones de estabilidad (ver proposición 3.3.4).
- Si $\alpha > \alpha_p$ el Sistema II es inestable dado que no se cumple la segunda condición de estabilidad.
- La condición para tener un canal con throughput positivo viene dada por $(1 + k_1/k)c > k_1$. Por lo cual podemos deducir que $L_0^{n_{tx}} < 1$. Por otra parte tenemos que $L_1^{n_p} < 0$ y que $L_1^{n_{tx}} > 1$.

Caso cuando $k_1v > \lambda$:

Si $k_1v > \lambda$ no habrá un polo con α en el intervalo $[0, 1]$ y \mathcal{C}^{sat} es un segmento de recta, como el segmento rojo que aparece en la Figura 3.8. Además podemos realizar los siguientes comentarios:

- Se verifica que $L_0^{n_{tx}} > 1$ y que $L_1^{n_p} < 0$.
- No se verifica la primera condición de estabilidad, por lo cual los puntos de equilibrio son inestables en todo el rango de α .

En esta sección avanzamos en la caracterización de los puntos de equilibrio del **Sistema II**. En función de $\text{sgn}(k_1v - \lambda)$ vimos que el conjunto de puntos de equilibrio tiene distinto comportamiento.

En un caso el conjunto es un conexo ubicado al exterior del triángulo factible y donde todos los puntos de equilibrio son inestables.

En el otro caso, los puntos de equilibrio están dentro o fuera del triángulo factible, según el valor de α . Un aspecto crucial es que mientras el punto de equilibrio esté en el semiplano positivo, es decir cuando $\alpha < \alpha_p$, el mismo es asintóticamente estable. Cuando $\alpha > \alpha_p$ entonces el punto de equilibrio se encuentra en el semiplano negativo, por fuera del triángulo factible, y es inestable.

En la sección 3.3.5 veremos qué implicancias tiene lo expresado anteriormente para la estabilidad del sistema conmutado.

Proposición 3.3.5. *El conjunto \mathcal{C}^{sat} compuesto por los puntos de equilibrio del Sistema II con $\alpha \in [0, 1]$ está incluido en una recta de ecuación $n_{tx} = \frac{\lambda+\mu}{k_1v-\lambda}n_p + \frac{c(1+k_1/k)v-\lambda}{k_1v-\lambda}$.*

Capítulo 3. Red IoT con múltiples capas de procesamiento

Demostración. Si fijamos $\dot{n}_p, \dot{n}_{tx} = 0$ en (3.14) y definimos $\beta = c(1 + k_1/k)v$ para simplificar la notación, obtenemos que:

$$\tilde{n}_{tx} = \frac{\beta}{k_1 v} - \frac{1 - \alpha}{\alpha} \frac{\mu}{k_1 v} \tilde{n}_p \quad (3.17)$$

$$\tilde{n}_p = \frac{\alpha \lambda (k_1 v - \beta)}{\alpha \lambda (k_1 v + \mu) + \mu (k_1 v - \lambda)} \quad (3.18)$$

Los puntos de equilibrio se pueden representar en el plano complejo como $\Gamma(\alpha) = \tilde{n}_p(\alpha) + j\tilde{n}_{tx}(\alpha)$, y al substituir esta expresión por (3.17) y (3.18) obtenemos:

$$\Gamma(\alpha) = \frac{\alpha [\lambda (k_1 v - \beta) + j\lambda (\beta + \mu)] + j\mu (\beta - \lambda)}{\alpha \lambda (k_1 v + \mu) + \mu (k_1 v - \lambda)}$$

Los puntos de equilibrio pueden expresarse como el mapeo de $\alpha \in \mathbb{R}$ por la transformación de Möbius $\Gamma(z) = \frac{az+b}{cz+d}$, con $a = \lambda(k_1 v - \beta) + j\lambda(\beta + \mu)$, $b = j\mu(\beta - \lambda)$, $c = \lambda(k_1 v + \mu)$ y $d = \mu(k_1 v - \lambda)$. Dado que $\text{Im}(c) = 0$ y $\text{Im}(d) = 0$, a partir de la proposición A.0.1 tenemos que los puntos de equilibrio están incluidos en una recta en el plano complejo. De forma análoga a la proposición 3.3.2, puede calcularse la ecuación de la recta y ésta queda como:

$$n_{tx} = \frac{\lambda + \mu}{k_1 v - \lambda} n_p + \frac{\beta - \lambda}{k_1 v - \lambda}$$

Si consideramos la función $\Gamma^{sat}(\alpha) = (\hat{n}_p^{sat}(\alpha), \hat{n}_{tx}^{sat}(\alpha))$, cuando $k_1 v \leq \lambda$ existe un factor de offloading $\alpha_p = \frac{\mu(\lambda - k_1 v)}{\lambda(k_1 v + \mu)}$ tal que $\Gamma^{sat}(\alpha_p) = \infty$ y \mathbf{C}^{sat} es disconexo. Si $k_1 v > \lambda$ entonces \mathbf{C}^{sat} es un segmento de recta ubicado en el exterior del triángulo factible (ver Figura 3.8). □

3.3.5. Análisis de estabilidad del sistema conmutado

Estamos ante un sistema conmutado en \mathbb{R}^2 que conmuta entre dos dinámicas lineales. La estabilidad de cada subsistema se puede estudiar mediante el análisis de los valores propios de las matrices fundamentales. Sin embargo, un sistema conmutado puede ser inestable aún cuando cada subsistema sea estable [30]. Esto nos lleva a realizar un análisis más cuidadoso del efecto de la conmutación. Si bien hay técnicas para analizar la estabilidad de sistemas conmutados [29, 31, 32], en este caso podemos explorar qué es lo que sucede en función de los parámetros del sistema.

Sabemos que el **Sistema I** sin restricciones de dominio es asintóticamente estable (proposición 3.3.3). Sin embargo el **Sistema II** es inestable si $k_1 v \geq \lambda + \mu$ o $\alpha \geq \frac{\mu(\lambda - k_1 v)}{\lambda(\mu + k_1 v)}$ (proposición 3.3.4).

Identificamos un total de seis escenarios diferentes en lo que refiere a estabilidad y ubicación de los puntos de equilibrio. Una revisión minuciosa de los diferentes escenarios variando los parámetros del sistema indica que todas las trayectorias del sistema conmutado converge a un único punto de equilibrio ubicado dentro del

3.3. Análisis del ODE determinístico

triángulo factible, ya sea \hat{n}^{lin} o \hat{n}^{sat} . Los escenarios según las distintas combinaciones de parámetros se muestran en la Figura 3.9 y en las Figuras B.1 a B.6 del Apéndice se muestran bosquejos de los escenarios.

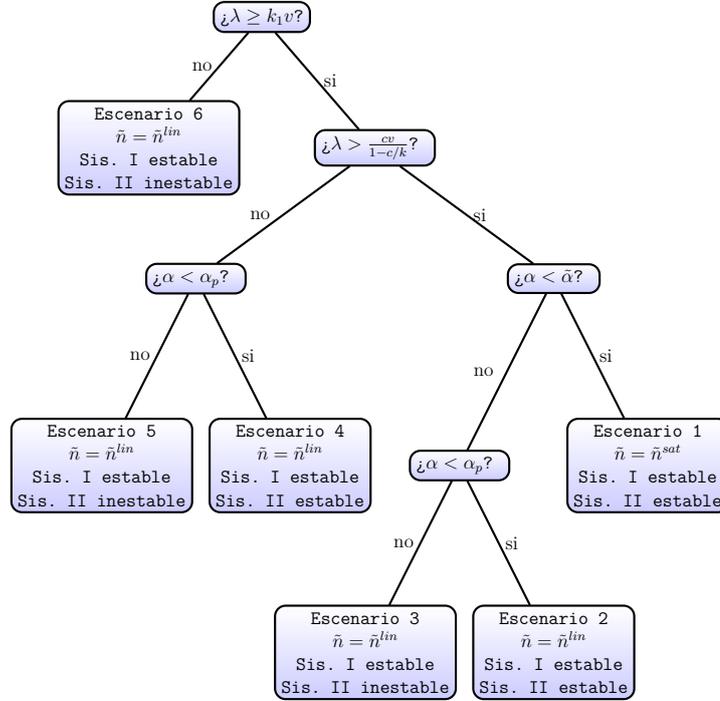


Figura 3.9: Distintos escenarios posibles según los parámetros del sistema

Como resultado general, en las condiciones donde el **Sistema II** es inestable, el punto de equilibrio de este subsistema se ubica en el exterior del triángulo factible, y el campo en $n_{tx} = c/k$ apunta hacia la parte inferior del triángulo. Por esta razón, todas las trayectorias iniciadas en la parte superior del triángulo cruzan hacia la parte inferior del triángulo y son atraídas por el equilibrio estable del **Sistema I** (ver Figura 3.11), por lo que el canal de comunicaciones no se encuentra saturado en el funcionamiento en régimen.

El escenario 1 se muestra en la Figura 3.10, en este caso el **Sistema II** es estable y los puntos de equilibrio de cada subsistema se encuentran ambos en el lado superior del triángulo. Este es el único escenario donde todas las trayectorias convergen al punto de equilibrio del **Sistema II**. Por lo tanto, el sistema alcanza un estado en régimen donde la cantidad de dispositivos transmitiendo saturan el canal de comunicaciones.

Con el nivel de conocimiento alcanzado hasta el momento, dados los parámetros del sistema, estamos en condiciones de determinar el punto de equilibrio del sistema. Esto es, la densidad de población de nodos procesando y transmitiendo en régimen. El punto de equilibrio repercute en la calidad de servicio, como será analizado en la próxima sección.

Capítulo 3. Red IoT con múltiples capas de procesamiento

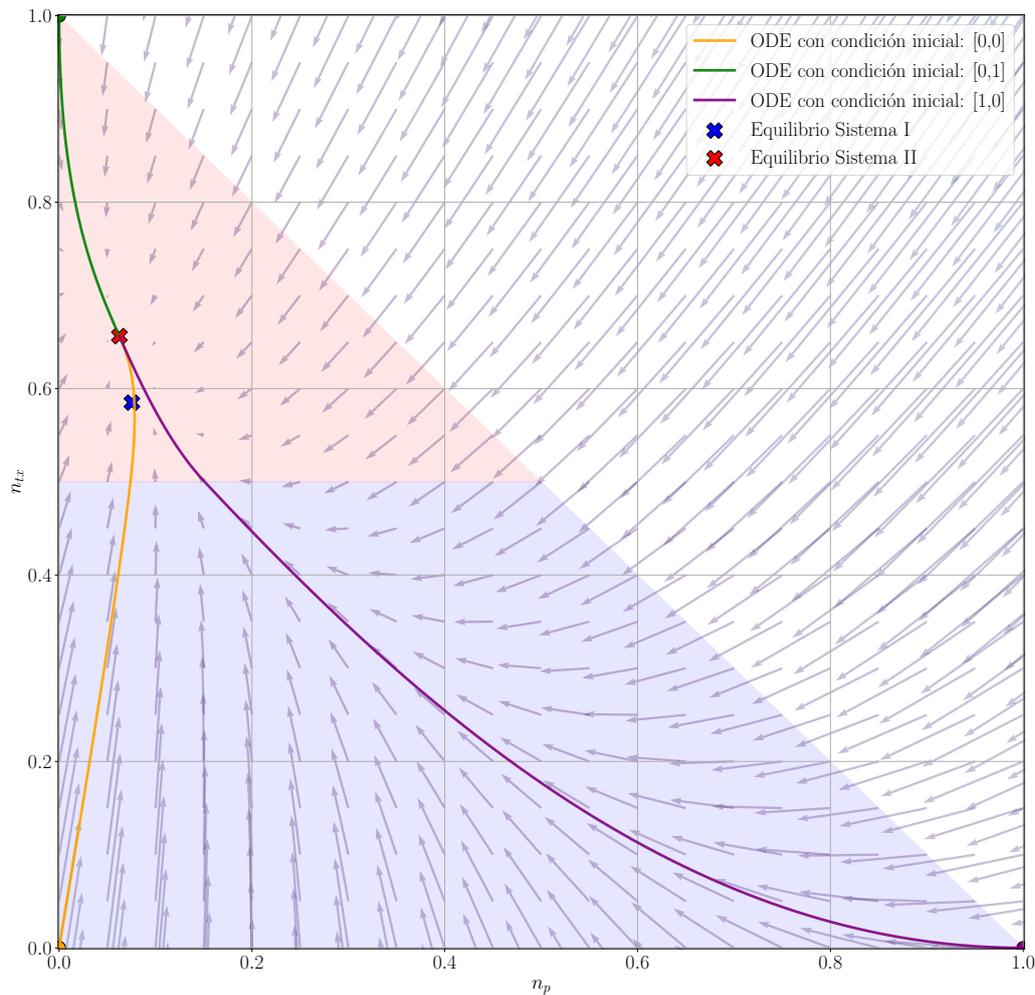


Figura 3.10: En este escenario, ambos puntos de equilibrio son estables y se encuentran ubicados en la parte superior del triángulo. Todas las trayectorias convergen al punto de equilibrio del **Sistema II**.

3.4. Calidad de Servicio

En esta sección se introducen dos métricas de calidad de servicio (QoS por sus siglas en inglés) que deben satisfacerse en el esquema de offloading: tasa de pérdidas de paquetes y tiempo de retardo.

3.4.1. Tasa de pérdida de paquetes

Es la tasa a la que arriban tareas cuando el dispositivo IoT se encuentra ocupado (procesando o transmitiendo). Asumiendo que todas las tareas transmitidas son luego procesadas, la tasa global de pérdidas está dada por:

3.4. Calidad de Servicio

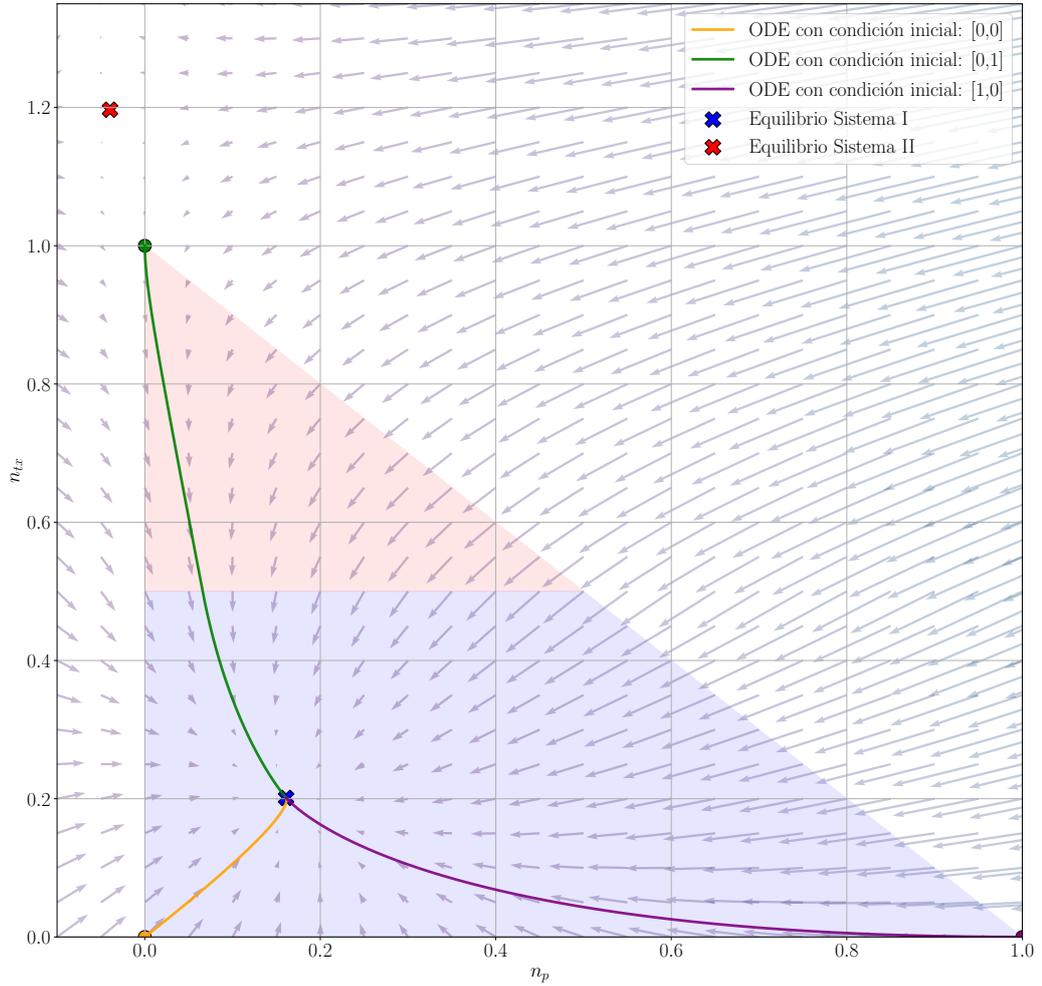


Figura 3.11: En este escenario el punto de equilibrio del **Sistema II** es inestable y se ubica fuera del triángulo factible, todas las trayectorias convergen al punto de equilibrio del **Sistema I**.

$$P_{\text{loss}}(\tilde{N}_p, \tilde{N}_{tx}) = \lambda(\tilde{N}_p + \tilde{N}_{tx}) \quad (3.19)$$

Podemos definir una tasa media de pérdidas por dispositivo como $p_{\text{loss}}(\tilde{N}_p, \tilde{N}_{tx}) = P_{\text{loss}}(\tilde{N}_p, \tilde{N}_{tx})/N$. Utilizando la aproximación determinística tenemos que $\mathbb{E}(p_{\text{loss}}) = \lambda(n_p + n_{tx})$. Una restricción de máxima tasa de pérdidas soportada de la forma $\mathbb{E}(p_{\text{loss}}) \leq \epsilon \lambda$ se verifica en el equilibrio $(\hat{n}_p, \hat{n}_{tx})$ si $\hat{n}_{tx} \leq \epsilon - \hat{n}_p$.

3.4.2. Tiempo de retardo

El tiempo para ejecutar una tarea procesada localmente tiene media $\tau_l = 1/\mu$. Se supone que los tiempos de procesamiento en Fog y Cloud son equivalentes y con valor $\tau_e = 1/\mu_e$. El tiempo de transmisión entre el dispositivo IoT y el gateway

Capítulo 3. Red IoT con múltiples capas de procesamiento

Fog tiene media $\tau_{tx} = \frac{1}{vC(n_{tx})/n_{tx}}$. El tiempo de transmisión entre el gateway Fog y los servidores Cloud se asumen muy inferiores a τ_e y τ_{tx} , y por esta razón se desprecian.

En un intervalo de tiempo ΔT se procesan en media $\mu N_p \Delta T$ tareas por segundo, y se transmiten $vC(N_{tx}) \Delta T$ tareas por segundo. La cantidad de tareas transmitidas y procesadas en el mismo lapso es por tanto $(\mu N_p + vC(n_{tx})N_{tx}) \Delta T$. por lo que podemos hacer una estimación del tiempo medio de retardo como:

$$\tilde{\tau} = \tau_l \frac{\mu N_p \Delta T}{(\mu N_p + vC(N_{tx})) \Delta T} + (\tau_{tx} + \tau_e) \frac{vC(N_{tx}) \Delta T}{(\mu N_p + vC(N_{tx})) \Delta T}$$

Si consideramos el límite fluido y el sistema en equilibrio, tenemos el valor medio del retardo como:

$$\mathbb{E}(\tau) = \tau_l \frac{\mu \hat{n}_p}{\mu \hat{n}_p + vC(\hat{n}_{tx})} + (\tau_{tx} + \tau_e) \frac{vC(\hat{n}_{tx})}{\mu \hat{n}_p + vC(\hat{n}_{tx})} \quad (3.20)$$

Podríamos estar interesados en imponer un tiempo de retardo máximo tal que $\mathbb{E}(\tau) \leq \tau_{max}$. Para cada subsistema, hay dos condiciones en las que el punto de equilibrio $(\hat{n}_p, \hat{n}_{tx})$ del sistema determina si esta condición puede satisfacerse o no.

Es decir, si tanto el tiempo medio de procesamiento local como el tiempo medio de procesarlo externamente satisfacen la restricción de retardo máximo, entonces cualquier proporción de nodos procesando/transmitiendo será satisfactoria. Y al revés, si ni el tiempo de procesamiento local ni el tiempo de procesamiento externo son menores que el tiempo de retardo máximo entonces ninguna proporción cumplirá la restricción.

Sistema I

Reemplazando la función de throughput en zona lineal en (3.20) obtenemos una región factible para $\mathbb{E}(\tau) \leq \tau_{max}$ dada por:

$$\hat{n}_{tx} \left(\frac{1}{kv} + \frac{1}{\mu_e} - \tau_{max} \right) \leq \frac{\mu}{kv} (\tau_{max} - 1/\mu) \hat{n}_p. \quad (3.21)$$

- Si $\frac{1}{\mu} \leq \tau_{max} < \frac{1}{kv} + \frac{1}{\mu_e}$ (Cond. 1), la región factible verifica:

$$\hat{n}_{tx} \leq \frac{\mu}{kv} \left(\frac{\tau_{max} - 1/\mu}{1/\mu_e + 1/kv - \tau_{max}} \right) \hat{n}_p = \kappa^{lin} \hat{n}_p.$$

- Si $\frac{1}{kv} + \frac{1}{\mu_e} < \tau_{max} \leq \frac{1}{\mu}$ (Cond. 2), la región factible verifica:

$$\hat{n}_{tx} \geq \kappa^{lin} \hat{n}_p.$$

Sistema II

Reemplazando la función de throughput en zona de saturación en (3.20) obtenemos una región factible para $\mathbb{E}(\tau) \leq \tau_{max}$ dada por:

$$\hat{n}_{tx}(\tau_{max} - 1/\mu_e + 1/k_1v)k_1v \leq (\tau_{max} - 1/\mu) (\mu\hat{n}_p + cv(1 + k_1/k))$$

- Si $\frac{1}{\mu} \leq \tau_{max} < \frac{1}{\mu_e} - \frac{1}{k_1v}$ (Cond. 3), la región factible verifica:

$$\hat{n}_{tx} \leq \frac{\mu}{k_1v} \left(\frac{\tau_{max} - 1/\mu}{\tau_{max} - 1/\mu_e + 1/k_1v} \right) \hat{n}_p + \quad (3.22)$$

$$\frac{cv(1 + k_1/k)(\tau_{max} - 1/\mu)}{k_1v(\tau_{max} - 1/\mu_e + 1/k_1v)} = \kappa^{sat}\hat{n}_p + \kappa_0 \quad (3.23)$$

- Si $\frac{1}{\mu_e} - \frac{1}{k_1v} < \tau_{max} \leq \frac{1}{\mu}$ (Cond. 4), la región factible verifica:

$$\hat{n}_{tx} \geq \kappa^{sat}\hat{n}_p + \kappa_0$$

3.5. Política óptima de offloading

Podemos formular un problema de optimización convexa para obtener el factor de offloading α que minimice una función de costo convexa $\Phi(n_p, n_{tx})$ con las restricciones de QoS presentadas en la sección anterior:

$$(P1) \quad \min_{n_p, n_{tx}} \quad \Phi(n_p, n_{tx}) \quad (3.24)$$

$$\text{subject to:} \quad 0 \leq n_{tx} \leq 1 \quad (3.25)$$

$$0 \leq n_p \leq 1 \quad (3.26)$$

$$n_p + n_{tx} \leq 1 \quad (3.27)$$

$$n_p + n_{tx} \leq \epsilon \quad (3.28)$$

$$\mathbb{E}(\tau^{lin}(n_p, n_{tx})) \leq \tau_{max} \quad (3.29)$$

$$\mathbb{E}(\tau^{sat}(n_p, n_{tx})) \leq \tau_{max} \quad (3.30)$$

Consideramos un escenario particular en donde el operador de la red IoT contrata unidades de procesamiento para computar tareas transmitidas por los dispositivos IoT. Los gateways Fog procesan todas las tareas transmitidas hasta una tasa máxima δN . El procesamiento adicional se deriva a los servidores Cloud que son más costosos.

Se considera una función de costo monetaria $\Phi(n_p, n_{tx}) = \Phi(n_{tx})$ con la siguiente expresión:

Capítulo 3. Red IoT con múltiples capas de procesamiento

$$\Phi(n_{tx}) = \begin{cases} \beta_1 n_{tx}, & \text{if } 0 \leq n_{tx} \leq \delta/kv \\ \beta_2 n_{tx} - (\beta_2 - \beta_1)\delta/kv, & \text{if } \delta/kv < n_{tx} \leq 1 \end{cases}$$

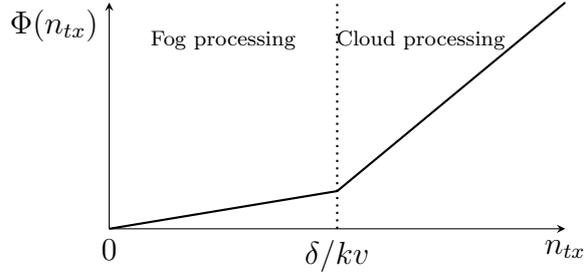


Figura 3.12: Costo de offloading hacia el fog y el cloud.

Los parámetros $\beta_1, \beta_2 > 0$ son los precios por computar en el Fog o el Cloud respectivamente. Asumimos que el procesamiento local no representa un costo monetario.

La función de costo decrece únicamente cuando n_{tx} decrece. Por lo tanto, una solución restringida a \mathcal{X}_{lin} es también una solución al problema (P1). Si no se encuentra solución dentro de \mathcal{X}_{lin} se puede luego buscar una solución en \mathcal{X}_{sat} . Por esta razón, se considera un nuevo problema (P2) que incluye la restricción \mathcal{X}_{lin} . Además, se agrega como restricción que la solución pertenezca a \mathbf{C}^{lin} , de forma de encontrar una solución que sea punto de equilibrio de la dinámica.

$$(P2) \quad \text{mín } t \quad (3.31)$$

$$\text{sujeto a:} \quad \beta_2 n_{tx} - (\beta_2 - \beta_1)\delta/kv \leq t \quad (3.32)$$

$$\beta_1 n_{tx} \leq t \quad (3.33)$$

$$n_{tx} \leq c/k \quad (3.34)$$

$$n_p + n_{tx} \leq 1 \quad (3.35)$$

$$-n_{tx} \leq 0 \quad (3.36)$$

$$n_p \leq 1 \quad (3.37)$$

$$-n_p \leq 0 \quad (3.38)$$

$$n_p + n_{tx} \leq \epsilon \quad (3.39)$$

$$\kappa n_p - n_{tx} \leq 0 \text{ or } -\kappa n_p + n_{tx} \leq 0 \quad (3.40)$$

$$n_{tx} = \frac{\lambda}{\lambda + kv} - \frac{\lambda + \mu}{\lambda + kv} n_p \quad (3.41)$$

$$(3.42)$$

Escribiendo $\mathbf{x} = [t, n_p, n_{tx}]^T$, el problema es equivalente al siguiente:

$$\min_{\mathbf{x}} \quad \mathbf{c}^T \mathbf{x} \quad (3.43)$$

$$\text{subject to:} \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \quad (3.44)$$

$$\mathbf{D}\mathbf{x} = \frac{\lambda}{\lambda + kv} \quad (3.45)$$

Con $\mathbf{c} = [1, 0, 0]$, $\mathbf{b} = [(\beta_2 - \beta_1)\delta/kv, 0, c/k, 0, 1, 0, \epsilon, 0]^T$, $\mathbf{D} = [0, \frac{\lambda+\mu}{\lambda+kv}, 1]$ y

$$\mathbf{A}^T = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & \pm\kappa \\ \beta_2 & \beta_1 & 1 & -1 & 0 & 0 & 1 & -1 \end{bmatrix}$$

En (P2), las restricciones (3.32)-(3.33) representan la función de costo lineal a tramos, (3.34)~(3.38) se requieren para restringir la solución al conjunto \mathcal{X}_{lin} , (3.39)-(3.40) son las restricciones de QoS y finalmente (3.41) se necesita para encontrar una solución que sea punto de equilibrio.

Dado que $\Gamma : \alpha \rightarrow (\hat{n}_p, \hat{n}_{tx})$ es una biyección, dada una solución $(\hat{n}_p^*, \hat{n}_{tx}^*)$ del problema convexo anterior, podemos obtener un único y óptimo factor de offloading α^* con la expresión:

$$\alpha^* = \frac{\mu \hat{n}_p^*}{\lambda(1 - \hat{n}_p^* - \hat{n}_{tx}^*)} \quad (3.46)$$

Si no se encuentra solución en (P2), debemos resolver un nuevo problema, análogo a (P2) y restringido a \mathcal{X}_{sat} , con las correspondientes restricciones de QoS y en \mathbf{C}^{sat} .

3.6. Resultados

Para el propósito de este estudio llevamos adelante una simulación de eventos discretos para recrear los procesos estocásticos que modelan la red IoT. Más información sobre el simulador implementado se encuentra en el Apéndice C). En la sección 3.6.1 presentamos los resultados obtenidos para un conjunto particular de parámetros para mostrar que el modelo de límite fluido es una muy buena aproximación a la dinámica del sistema simulado. Luego, en la sección 3.6.2 calculamos el factor óptimo de offloading de un problema de optimización con restricciones de QoS.

3.6.1. Aproximación de límite fluido

Un conjunto de parámetros interesante para simular es el del escenario representado en la Figura 3.10. En este caso teníamos que dependiendo del factor de offloading podemos tener un punto de equilibrio en donde el canal se encuentre saturado o no.

Capítulo 3. Red IoT con múltiples capas de procesamiento

| Parámetro | Descripción | Valor | Unidad |
|-----------------|---|--------------------|--------------|
| N | Cantidad de dispositivos IoT | 500 | - |
| M | Cantidad de gateways Fog | 25 | - |
| λ | Tasa de arribo de tareas | 1 | tareas/s |
| μ | Tasa de procesamiento de tareas | 0.5 | tareas/s |
| c_g | Throughput máximo por gateway | 100 | Mbps |
| k | Throughput en zona lineal | 10 | Mbps/usuario |
| k_1 | Parámetro de saturación de throughput | 3 | - |
| v | Factor tareas por bit | 2×10^{-7} | tareas/bit |
| $N_{tx}(t=0)$ | Cantidad inicial de nodos transmitiendo | 0 | - |
| $N_p(t=0)$ | Cantidad inicial de nodos procesando | 0 | - |
| $N_{idle}(t=0)$ | Cantidad inicial de nodos en espera | 500 | - |
| T | Tiempo de simulación | 50 | s |

Tabla 3.2: Parámetros de la simulación 3.6.1

Realizamos dos simulaciones con distintos factores de offloading: $\alpha = 0.1$ y $\alpha = 0.9$, los demás parámetros de la simulación son iguales para ambos experimentos y se detallan en la Tabla 3.2. Los parámetros fueron escogidos de forma que cuando $\alpha = 0.1$ estemos en el escenario 3 y cuando $\alpha = 0.9$ estemos en el escenario 1, el lector puede repasar los escenarios en la Figura 3.9.

Los resultados obtenidos se muestran en la Figura 3.13. Podemos observar que para cada densidad de población (transmitiendo, procesando y en espera), la solución del ODE determinístico es una muy buena aproximación de las densidades simuladas, no solamente respecto al punto de equilibrio sino también de la respuesta transitoria.

En un primer caso con $\alpha = 0.9$ el punto de equilibrio pertenece a \mathcal{X}_{lin} y la trayectoria nunca entra en \mathcal{X}_{sat} , por lo que no existe conmutación entre subsistemas. Sin embargo, con $\alpha = 0.1$, la trayectoria inicia en \mathcal{X}_{lin} pero el punto de equilibrio está en \mathcal{X}_{sat} por lo que se registra una conmutación entre subsistemas cuando la densidad de nodos transmitiendo alcanza $c/k = 0.5$.

De forma de comparar numéricamente el punto de equilibrio del modelo determinístico con el valor medio del proceso estocástico y a los efectos de reducir el efecto de la respuesta transitoria, realizamos una nueva simulación modificando la condición inicial para que coincida con el punto de equilibrio esperado. En el caso con $\alpha = 0.9$ fijamos entonces $(N_{tx}(0), N_p(0)) = (N\hat{n}_p^{lin}, N\hat{n}_{tx}^{lin})$ y para el caso con $\alpha = 0.1$ fijamos $(N_{tx}(0), N_p(0)) = (N\hat{n}_p^{sat}, N\hat{n}_{tx}^{sat})$. Los resultados se muestran en la Tabla 3.3. Para cada clase y factor de offloading comparamos el valor medio del proceso estocástico con el punto de equilibrio del modelo fluido. Aún con un número finito de dispositivos Iot ($N = 500$) podemos concluir que el modelo fluido es una muy buena aproximación para obtener el valor esperado del proceso estocástico.

| Parámetro | $\alpha = 0.1$ | | $\alpha = 0.9$ | |
|-----------|----------------|---------------|----------------|---------------|
| | Media Sim. | Modelo fluido | Media Sim. | Modelo fluido |
| n_{tx} | 0.8947 | 0.8985 | 0.1503 | 0.1515 |
| n_p | 0.0172 | 0.0169 | 0.5512 | 0.5454 |

Tabla 3.3: Valor medio del proceso estocástico vs punto de equilibrio del modelo fluido

| Parámetro | Descripción | Valor | Unidad |
|-------------------|---|-----------|------------------|
| N | Cantidad de dispositivos IoT | 1000 | - |
| M | Cantidad de gateways Fog | 50 | - |
| λ | Tasa de arribo de tareas | 3 | tareas/s |
| μ | Tasa de procesamiento de tareas (local) | 2.8 | tareas/s |
| μ_e | Tasa de procesamiento de tareas (Fog or Cloud) | 40 | tareas/s |
| c_g | Throughput máximo por gateway | 1000 | Mbps |
| k | Throughput en zona lineal zone | 100 | Mbps |
| k_1 | Parámetro de saturación de throughput parameter | 30 | - |
| v | Factor tareas por bit | 10^{-5} | tareas/bit |
| δ | Tasa máxima de procesamiento en gateway Fog | 0.03 | tareas/s |
| β_1 | Precio de procesamiento en Fog | 0.5 | Unidad Monetaria |
| β_2 | Precio de procesamiento en Cloud | 4 | Unidad Monetaria |
| ϵ_1 | Tasa Máxima de Pérdidas (Exp. I) | 10 | % |
| ϵ_2 | Tasa Máxima de Pérdidas (Exp. II) | 4 | % |
| τ_{max} | Tiempo de Retardo Máximo | 50 | ms |
| $N_{tx}(t = 0)$ | Número inicial de nodos transmitiendo | 0 | - |
| $N_p(t = 0)$ | Número inicial de nodos procesando | 0 | - |
| $N_{idle}(t = 0)$ | Número inicial de nodos en espera | 1000 | - |
| T | Tiempo de simulación | 20 | s |

Tabla 3.4: Parámetros de la simulación 3.6.2

3.6.2. Factor óptimo de offloading

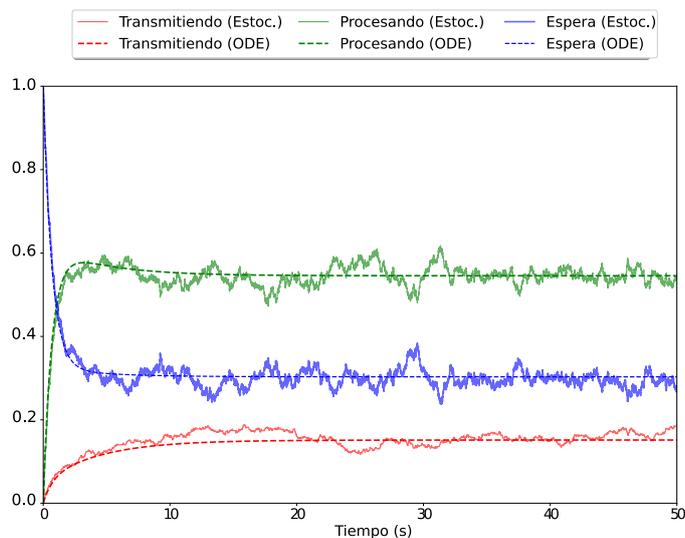
En esta parte simulamos un total de $N = 1000$ dispositivos IoT y $M = 50$ gateways Fog. Realizamos dos experimentos con diferentes valores de Tasa Máxima de Pérdidas. En el **Experimento I** se fija $\epsilon = 10\%$ y para el **Experimento II** se fija $\epsilon = 4\%$. Los restantes parámetros son iguales en ambos experimentos y se detallan en la Tabla 3.4.

El problema de optimización se resuelve mediante CVXPY [33] y los resultados se incluyen en la Tabla 3.5. Para ilustrar mejor estos resultados, en la Figura 3.14 se muestran las simulaciones del proceso estocástico, junto con las trayectorias del

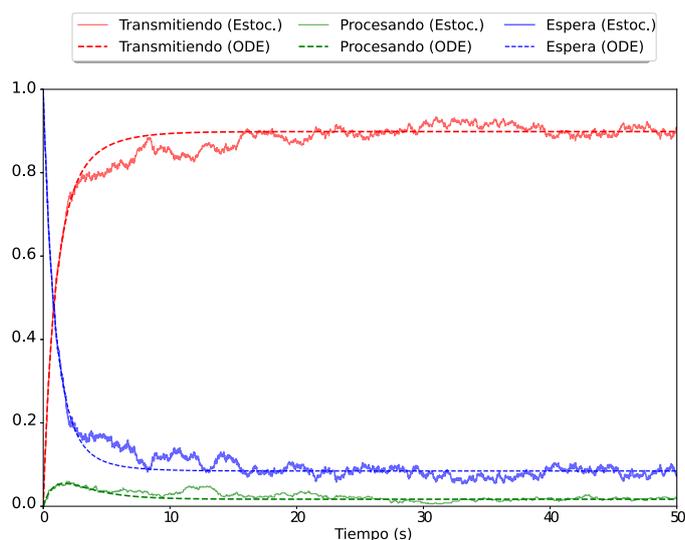
| Parámetro | Experimento I | Experimento II |
|------------------------|---------------|----------------|
| \hat{n}_p^* | 0.0463 | 0.0115 |
| \hat{n}_{tx}^* | 0.0265 | 0.0285 |
| α^* | 0.0466 | 0.0112 |
| $\Phi(\hat{n}_{tx}^*)$ | 0.1050 | 0.1128 |

Tabla 3.5: Soluciones del problema de optimización

Capítulo 3. Red IoT con múltiples capas de procesamiento



(a) $\alpha = 0.9$

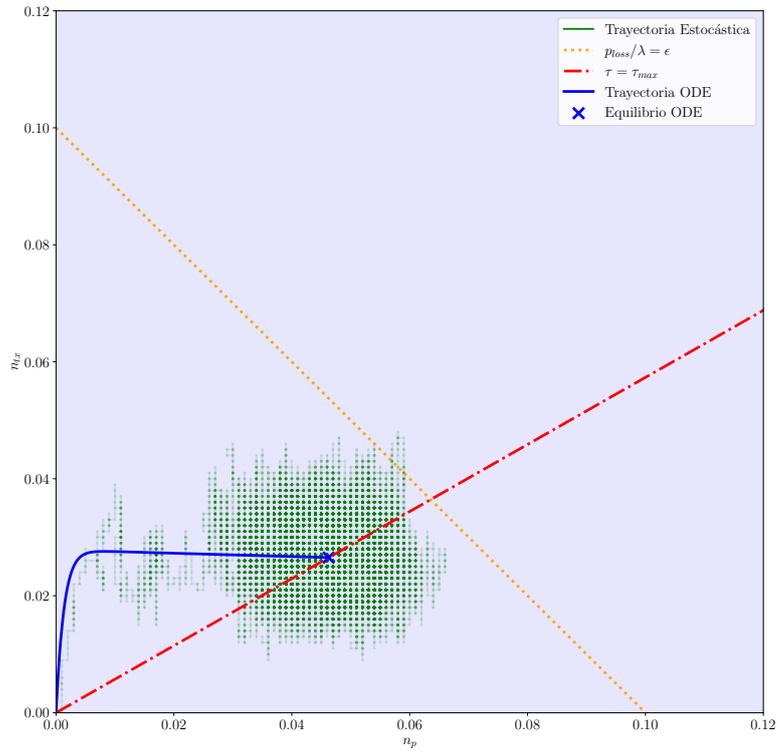


(b) $\alpha = 0.1$

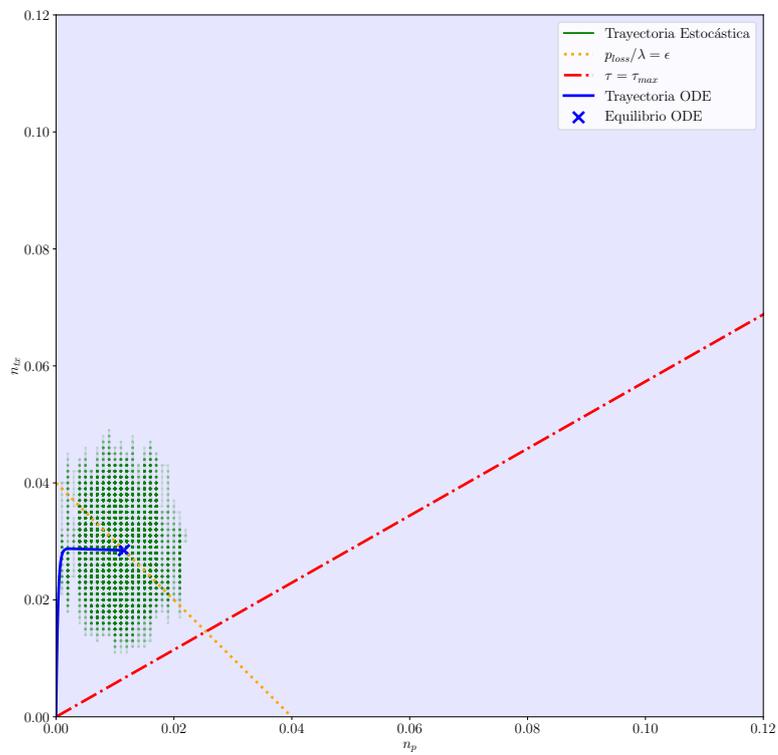
Figura 3.13: Evolución temporal de las densidades de población. Las trayectorias estocásticas de cada clase se compara con la correspondiente aproximación de límite fluido. Se muestran dos simulaciones con valores distintos de factor de offloading.

ODE determinístico, usando los dos factores de offloading obtenidos.

En la Tabla 3.6 se muestra información importante respecto a las restricciones del problema de optimización. Bajo el primer experimento, el multiplicador dual asociado a la eq. (3.40) es no nulo, lo que significa que la restricción de retardo máximo se activa. En cuanto a la función de costo, el multiplicador asociado a la eq. (3.32) es también no nulo por lo que el procesamiento realizado en el fog no



(a) Experimento I



(b) Experimento II

Figura 3.14: Diagrama de fase con las trayectorias del ODE, la simulación estocástica y las restricciones de QoS.

Capítulo 3. Red IoT con múltiples capas de procesamiento

| Experimento | Multiplicadores de Lagrange (Eqs. 3.32 ~ 3.40) |
|-------------|--|
| I | [1.0, 0, 0, 0, 0, 0, 0, 0, 0.36] |
| II | [1.0, 0, 0, 0, 0, 0, 0, 0.24, 0] |

Tabla 3.6: Multiplicadores de Lagrange del problema de optimización

puede satisfacer toda la demanda y se requieren servidores Cloud. Esto es consistente con el hecho que $\hat{n}_{tx} > \delta/kv$. En el segundo experimento, el multiplicador asociado a eq. (3.39) es no nulo, lo que significa que la restricción de tasa de pérdidas se activa, sin embargo la restricción de retardo máximo no se activa. También en este caso se requiere procesar tareas en el Cloud.

Las métricas de QoS para ambos experimentos se presentan en la Tabla 3.5 y en la Figura 3.16. En la Figura 3.15 se muestra la evolución temporal de las métricas de QoS para el Experimento I.

Bajo el Experimento I, el Tiempo de Retardo medio viola el Tiempo de Retardo Máximo $\tau_{max} = 50\text{ms}$. Esto no es inesperado considerando el hecho que esta es solo una realización del proceso estocástico con un conjunto finito de dispositivos IoT, y el problema de optimización está basado en el límite asintótico del Proceso de Markov.

| Parámetro | Experimento I | | Experimento II | |
|------------------|---------------|-----------------------|----------------|-----------------------|
| | Media | Dev. Estándar | Media | Dev. Estándar |
| Retardo (s) | 0.0501 | 2.03×10^{-3} | 0.0385 | 9.84×10^{-4} |
| Tasa de pérdidas | 0.0727 | 7.54×10^{-3} | 0.0407 | 4.59×10^{-3} |

Tabla 3.7: Resultados de QoS

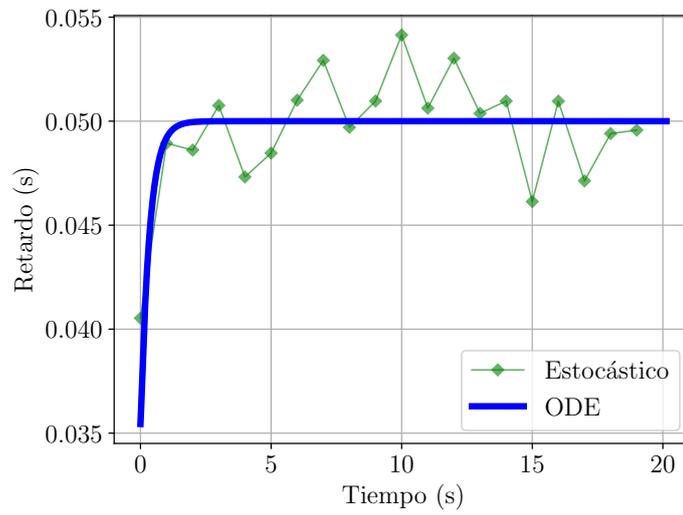
Si se requiere una optimización robusta de la realización estocástica, trabajos futuros deberán fijar restricciones más ajustadas calculando por ejemplo elipses de confianza.

3.7. Conclusiones y Trabajo Futuro

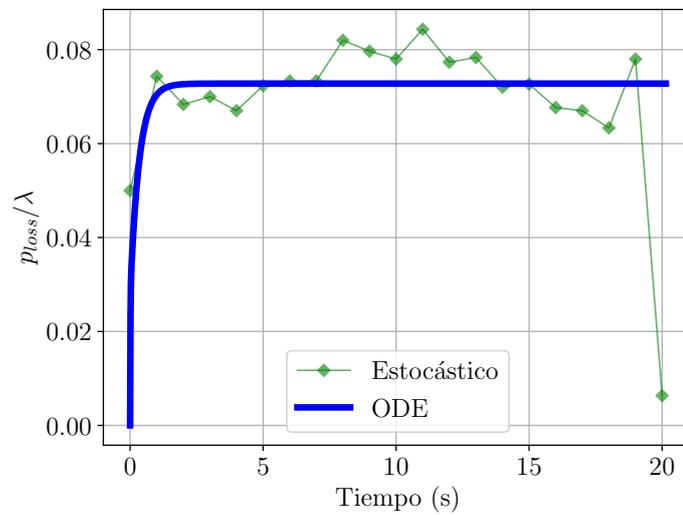
Este trabajo propone un enfoque novedoso para analizar una red IoT de gran escala a través de la aproximación fluida del Proceso de Markov que describe la dinámica del sistema. Los resultados muestran que el modelo determinístico es una excelente aproximación del proceso estocástico. En particular, el modelo logra capturar el punto de equilibrio de la red IoT. El conocer la ubicación del punto de equilibrio en función de los parámetros del problema, permite construir un problema de optimización convexo para hallar el factor de offloading óptimo respetando métricas de QoS.

En estudios futuros se puede investigar cómo realizar una optimización robusta cuando la cantidad de dispositivos IoT es un número finito considerando intervalos de confianza.

3.7. Conclusiones y Trabajo Futuro



(a) Tiempo de retardo



(b) Tasa de pérdidas

Figura 3.15: Evolución de la calidad de servicio en el Experimento I

Capítulo 4

Red IoT con Energy Harvesting

Los *Edge Devices* (EDs), sean dispositivos IoT o equipos móviles, requieren para su funcionamiento de una fuente de energía. En la medida que no se disponga de una alimentación de la red eléctrica es necesario asegurar energía mediante unidades de almacenamiento. Ya sea con baterías electroquímicas previamente cargadas y/o mediante la cosecha de energía del ambiente (en inglés *energy harvesting*).

La batería de tipo electroquímica presenta distintos problemas como son una vida útil limitada, su descarga aunque no esté en uso así como el impacto ambiental de los metales y químicos utilizados.

El *energy harvesting* busca superar algunos de estos desafíos. Existen distintas tecnologías de harvesting en función del origen de la energía, como pueden ser solar, eólica, hidráulica, RF, vibraciones mecánicas, calor humano, etc.

Los sistemas de cosecha de energía pueden ser de tipo Cosecha-Uso o de Cosecha-Almacenamiento-Uso, la diferencia en el segundo caso es la existencia de una unidad que almacena la energía cosechada, a través de baterías recargables o supercondensadores. De esta manera se pueden aprovechar las fuentes intermitentes de energía en los momentos donde estas no se encuentran presentes.

El *energy harvesting* puede analizarse mediante modelos determinísticos o estocásticos. Para los primeros es preciso contar con una correcta predicción de la llegada de energía y sus fluctuaciones. En general los modelos estocásticos son más apropiados.

La importancia del modelado radica en la posibilidad de ajustar políticas de sensado, transmisión y procesamiento para maximizar la vida útil de una red de EDs o bien para tener acotadas las pérdidas de tareas por falta de energía mientras se optimiza cierto objetivo del sistema en su conjunto.

4.1. Modelo del sistema

Extenderemos el modelo analizado en el capítulo 3 e introduciremos un modelo de energía. Cada dispositivo IoT cuenta con una batería que almacena valores discretos de energía: $\{0, 1, 2, \dots, E\}$ siendo E la capacidad máxima.

Sea $E_m(t)$ el nivel de energía de la batería del m -ésimo dispositivo IoT en

Capítulo 4. Red IoT con Energy Harvesting

tiempo t , con $m = 1, 2, \dots, N$. Suponemos que el dispositivo tiene la posibilidad de recolectar energía del ambiente (ya sea solar, viento, radiofrecuencia). En el modelo solo suponemos que el arribo de estas unidades discretas de energía sigue un proceso Poisson con tasa γ .

Procesar localmente o transmitir una tarea consume una unidad de energía y el descuento se efectiviza cuando el dispositivo pasa al estado en espera. A diferencia del modelo analizado en el capítulo anterior, en este caso asumimos que el canal de comunicaciones se encuentra siempre en zona lineal.

A modo de ejemplo, consideremos la siguiente secuencia (*Estado, Energía*) de un dispositivo en espera que luego comienza a transmitir:

$$(Idle, E_m(t_1) = 5) \longrightarrow (Tx, E_m(t_2) = 5) \longrightarrow (Idle, E_m(t_3) = 4),$$

o de un dispositivo en espera que luego comienza a procesar:

$$(Idle, E_m(t'_1) = 1) \longrightarrow (P, E_m(t'_2) = 1) \longrightarrow (Idle, E_m(t'_3) = 0).$$

Si llega una unidad de energía a un nodo completamente cargado ($E_m(t) = E$), la unidad se descarta. Si llega una tarea a un nodo en espera con batería descargada ($E_m(t) = 0$) la tarea se descarta por falta de energía. Es importante remarcar que con este modelo no pueden haber nodos descargados que se encuentren procesando o transmitiendo.

Siguiendo la línea del Capítulo 3, definimos $\tilde{N}_p^j(t)$, $\tilde{N}_{tx}^j(t)$, $\tilde{N}_{id}^j(t)$ como procesos de conteo con la cantidad de nodos con nivel de energía j que se encuentran procesando, transmitiendo o en estado idle respectivamente. De los procesos con $j = 0$ solo nos interesa la clase de nodos en espera. Esto es porque no pueden existir nodos que se encuentren transmitiendo o procesando sin energía disponible.

Nuestro espacio de estados S está dado entonces por:

$$S = \{(Y_{id}^0, Y_{id}^1, Y_p^1, Y_{tx}^1, Y_{id}^2, Y_p^2, Y_{tx}^2, \dots, Y_{id}^E, Y_p^E, Y_{tx}^E) \in \mathbb{N}^{3 \times E + 1} | \\ 0 \leq Y_{id}^{j=0,1,\dots,E} \leq N, 0 \leq Y_p^{j=1,2,\dots,E} \leq N, 0 \leq Y_{tx}^{j=1,2,\dots,E} \leq N, \\ \sum_{j=0}^E Y_{id}^j + \sum_{j=1}^E Y_p^j + \sum_{j=1}^E Y_{tx}^j = N\}$$

Consideramos el proceso estocástico $(\tilde{Z}^N)_t$ con espacio de estados S que representa la población de una red IoT con la dinámica presentada previamente. Es decir tenemos que:

$$(\tilde{Z}^N)_t = (\tilde{N}_{id}^0, \tilde{N}_{id}^1, \tilde{N}_p^1, \tilde{N}_{tx}^1, \tilde{N}_{id}^2, \tilde{N}_p^2, \tilde{N}_{tx}^2, \dots, \tilde{N}_{id}^E, \tilde{N}_p^E, \tilde{N}_{tx}^E)_t \in \mathbb{N}^{3 \times E + 1}$$

Sean $e_p^j, e_{tx}^j, e_{id}^j$ vectores de dimensión $3 \times E + 1$ que valen cero en todas sus entradas salvo en la entrada correspondiente a su clase con nivel de batería j y en cuyo caso vale 1.

4.1. Modelo del sistema

Consideremos un cierto estado $s \in S$:

$$s^T = (N_{id}^0, N_{id}^1, N_p^1, N_{tx}^1, N_{id}^2, N_p^2, N_{tx}^2, \dots, N_{id}^E, N_p^E, N_{tx}^E) \in \mathbb{N}^{3 \times E + 1}$$

Nos interesa calcular la intensidad de las transiciones posibles $\tilde{q}(s, s + v)$ desde el estado s hasta el estado $s + v$, $v \in \mathbb{N}^{3 \times E + 1}$.

La dirección del salto v pertenece a un conjunto bastante acotado debido a la naturaleza de nuestro problema. En primer lugar cada población puede incrementarse o decrementarse en una unidad. Por otra parte, para cada evento, a lo sumo habrán dos coordenadas del estado que sufrirán cambios.

A modo de ejemplo, si una red se encuentra en el estado s y llega una unidad de energía a un nodo de la clase tx con nivel de energía j , entonces el nuevo estado del sistema será $s' = s + v$ con $v = e_{tx}^{j+1} - e_{tx}^j$. En otro caso, cuando una tarea culmina de procesarse en un nodo con nivel de energía j , el estado se moverá en la dirección $v' = e_{id}^{j-1} - e_{id}^j$.

Ahora queremos saber cuál es la intensidad de los distintos saltos, es decir $\tilde{q}(s, s + v)$ para cualquier dirección v posible según nuestro problema.

Comencemos analizando los movimientos que involucran a la población de nodos en espera con batería baja ($j = 0, j = 1$). La población de nodos en espera sin energía puede crecer únicamente luego que terminen de procesarse o transmitirse tareas desde nodos con nivel de energía $j = 1$, por lo que el incremento de la coordenada N_{id}^0 responde a un salto con alguna de estas dos direcciones: $e_{id}^0 - e_p^1$ y $e_{id}^0 - e_{tx}^1$.

Veamos otro ejemplo que involucra la carga de energía. Si llega una unidad de energía a un nodo que se encuentre procesando con nivel de energía $j = 4$, entonces tendremos un salto del proceso en la dirección $v = e_p^5 - e_p^4$.

La intensidad de las transiciones en estas direcciones dependerá de la cantidad de dispositivos de la clase desde donde se transiciona y la correspondiente tasa de procesamiento μ , transmisión kv o cosecha de energía γ . La intensidad de transiciones desde el estado s hacia el estado $s + v$ la representamos como $\tilde{q}(s, s + v)$, si por ejemplo $v' = e_{id}^0 - e_p^1$ entonces $\tilde{q}(s, s + v') = \mu N_p^1$.

En las Tablas 4.1, 4.2 y 4.3 se detallan todas las direcciones de salto posibles y sus correspondientes intensidades. A los efectos de tener bien ordenadas las direcciones de los saltos, y no perdernos de ninguna, optamos por hacer un agrupamiento según la coordenada que se incremente. Es por esto que el primer término de la dirección es siempre el que corresponde a la coordenada que esté siendo considerada.

Estamos en condiciones de calcular el *process drift* que se define como:

$$Q(s) = \sum_v v \tilde{q}(s, s + v)$$

Si sumamos todas las direcciones de los saltos ponderadas por la respectiva intensidad tenemos que:

Capítulo 4. Red IoT con Energy Harvesting

| Incremento | Dirección v | $\tilde{q}(s, s + v)$ |
|------------|-----------------------|---------------------------------|
| N_{id}^0 | $e_{id}^0 - e_p^1$ | μN_p^1 |
| | $e_{id}^0 - e_{tx}^1$ | $kv N_{tx}^1$ |
| N_{id}^1 | $e_{id}^1 - e_{id}^0$ | γN_{id}^0 |
| | $e_{id}^1 - e_p^2$ | μN_p^2 |
| | $e_{id}^1 - e_{tx}^2$ | $kv N_{tx}^2$ |
| N_p^1 | $e_p^1 - e_{id}^1$ | $\alpha \lambda N_{id}^1$ |
| N_{tx}^1 | $e_{tx}^1 - e_{id}^1$ | $(1 - \alpha) \lambda N_{id}^1$ |

Tabla 4.1: Intensidades de nodos con carga baja

| Incremento | Dirección v | $\tilde{q}(s, s + v)$ |
|------------|---------------------------|---------------------------------|
| N_{id}^j | $e_{id}^j - e_{id}^{j-1}$ | γN_{id}^{j-1} |
| | $e_{id}^j - e_p^{j+1}$ | μN_p^{j+1} |
| | $e_{id}^j - e_{tx}^{j+1}$ | $kv N_{tx}^{j+1}$ |
| N_p^j | $e_p^j - e_p^{j-1}$ | γN_p^{j-1} |
| | $e_p^j - e_{id}^j$ | $\alpha \lambda N_{id}^j$ |
| N_{tx}^j | $e_{tx}^j - e_{tx}^{j-1}$ | γN_{tx}^{j-1} |
| | $e_{tx}^j - e_{id}^j$ | $(1 - \alpha) \lambda N_{id}^j$ |

Tabla 4.2: Intensidades de nodos con carga intermedia ($j = 2, \dots, E - 1$)

$$\tilde{Q}^N(s) = e_{id}^0 (\mu N_p^1 + kv N_{tx}^1 - \gamma N_{id}^0) + e_{id}^1 (\gamma N_{id}^0 + \mu N_p^2 + kv N_{tx}^2 - \alpha \lambda N_{id}^1 - (1 - \alpha) \lambda N_{id}^1 - \gamma N_{id}^1) + \dots + e_{tx}^E ((1 - \alpha) \lambda N_{id}^E + \gamma N_{tx}^{E-1})$$

Es posible escribir el *drift* en notación matricial como:

$$\tilde{Q}^N(s) = F \cdot s \quad (4.1)$$

A modo de ejemplo, si tomamos el caso donde $E = 3$ se obtiene la siguiente matriz $F \in \mathcal{M}_{10 \times 10}$:

4.1. Modelo del sistema

| Incremento | Dirección v | $\tilde{q}(s, s + v)$ |
|------------|---------------------------|---------------------------------|
| N_{id}^E | $e_{id}^E - e_{id}^{E-1}$ | γN_{id}^{E-1} |
| N_p^E | $e_{id}^E - e_{id}^{E-1}$ | γN_{id}^{E-1} |
| | $e_p^E - e_{id}^E$ | $\alpha \lambda N_{id}^E$ |
| N_{tx}^E | $e_{tx}^E - e_{tx}^{E-1}$ | γN_{tx}^{E-1} |
| | $e_{tx}^E - e_{id}^E$ | $(1 - \alpha) \lambda N_{id}^E$ |

Tabla 4.3: Intensidades de nodos completamente cargados

$$F = \begin{bmatrix} -\gamma & 0 & \mu & kv & 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma & -\gamma - \lambda & 0 & 0 & 0 & \mu & kv & 0 & 0 & 0 \\ 0 & \alpha \lambda & -\gamma - \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1 - \alpha) \lambda & 0 & -\gamma - kv & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 & -\gamma - \lambda & 0 & 0 & 0 & \mu & kv \\ 0 & 0 & \gamma & 0 & \alpha \lambda & -\gamma - \mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & (1 - \alpha) \lambda & 0 & -\gamma - kv & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & 0 & \alpha \lambda & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma & (1 - \alpha) \lambda & 0 & -kv \end{bmatrix} \quad (4.2)$$

4.1.1. Aproximación fluida

Con un procedimiento análogo al realizado en la proposición 3.2.1 puede verse que el proceso estocástico \tilde{Z}_t^N puede someterse a un límite fluido para aproximar el proceso escalado a un ODE determinístico.

Definamos en primer lugar $Q(l \in S) = Fl$, donde F está dada por (4.1). Se puede verificar que $\frac{\tilde{Q}^N(l)}{N} = Q(l/N)$ y por lo tanto:

$$\lim_{N \rightarrow \infty} \sup_{l \in S} \left\| \frac{\tilde{Q}^N(l)}{N} - Q(l/N) \right\| = 0, \quad (4.3)$$

Dado que se cumple 4.3 y que $Q(\cdot)$ es Lipschitz, entonces por el Teorema 2.1 pág. 456 en [11] el proceso escalado $(Z_t) = \left(\frac{1}{N} \tilde{Z}_t^N\right)$ cuando $N \rightarrow \infty$ converge en probabilidad a $z(t)$, solución del siguiente ODE determinístico:

$$\dot{z} = Q(z(t)) = Fz(t), \quad z(t_0) = z_0 \quad (4.4)$$

4.2. Modelo determinístico

Estudiaremos ahora el sistema de ecuaciones diferenciales definido en (4.4) que surge de tomar el límite de la aproximación fluida. Consideraremos un vector de estados z de la siguiente forma:

$$z = \left[n_{id}^0, n_{id}^1, n_p^1, n_{tx}^1, n_{id}^2, n_p^2, n_{tx}^2, \dots, n_{id}^E, n_p^E, n_{tx}^E \right]^T \in \mathbb{N}^{3 \times E + 1}$$

Para facilitar la notación nos referiremos en ocasiones a la coordenada i -ésima de $z(t)$ como $z_i(t)$, con $i = 1, 2, \dots, 3 \times E + 1$.

Una primera observación que se puede realizar respecto a la matriz F es que cada una de sus columnas suman cero, de donde:

$$\mathbb{1} \cdot F = \mathbf{0} = 0 \cdot \mathbb{1}. \quad (4.5)$$

De esta manera, F tiene vector propio izquierdo $\mathbb{1}$ con valor propio asociado $\lambda = 0$. Por esta razón la dinámica 4.4 tendrá infinitos puntos de equilibrio.

Una segunda observación es que se verifica que $\mathbb{1} \cdot [F \cdot z(t)] = \mathbb{1} \cdot \dot{z}(t) = 0 \quad \forall t$. Esto quiere decir que la cantidad $\sum_i z_i(t)$ se mantiene constante durante toda la trayectoria y queda definida por la condición inicial $\sum_i z_i(t = 0) \quad \forall t$.

Además, y dado que F es Metzler, por el teorema 2.4.3 el sistema definido por 4.4 es positivo. Se verifica entonces que cada coordenada del vector de estados es no negativa, es decir $z_i(t) \geq 0 \quad \forall(i, t)$. Como vimos anteriormente $\sum_i z_i(t) = \sum_i z_i(t = 0)$ por lo cual si tenemos por condición inicial que $\mathbb{1}z(t = 0) = 1$, entonces $0 \leq z_i(t) \leq 1 \quad \forall(i, t)$. Podemos concluir entonces que el n -símplex¹ de dimensión $3 \times E$ es invariante frente a la dinámica (4.4).

En la siguiente sección formularemos un nuevo ODE, muy similar al anterior, que incluye la restricción $\mathbb{1}z = 1$ lo cual nos permitirá establecer un único punto de equilibrio.

4.2.1. ODE considerando suma constante en la variable de estados

La suma de los estados se mantiene constante entonces podemos eliminar una variable en función de las demás. Elegimos eliminar los nodos en espera sin energía y pasar a un sistema de dimensión $3 \times E$. Vamos a considerar la siguiente condición inicial:

¹Un n -símplex en \mathbb{R}^{n+1} es el conjunto $\{(x_0, \dots, x_n) \in \mathbb{R}^{n+1} \mid \sum_i x_i = 1 \text{ y } x_i \geq 0 \quad \forall i\}$. Un n -símplex es el análogo en n dimensiones de un triángulo.

4.3. Simulación del sistema

$$\sum_i z_i(t=0) = 1. \quad (4.6)$$

Dado que nuestra variable de estados es una densidad, se cumple que:

$$n_{id}^0 = 1 - \sum_{j=1}^E n_{id}^j - \sum_{j=1}^E n_p^j - \sum_{j=1}^E n_{tx}^j \quad (4.7)$$

Por otro lado \dot{n}_{id}^1 se puede obtener multiplicando la segunda fila de la matriz F con el vector de estados y nos queda:

$$\dot{n}_{id}^1 = \gamma n_{id}^0 - (\gamma + \lambda)n_{id}^1 + \mu n_p^2 + kv n_{tx}^2 \quad (4.8)$$

Combinando las ecuaciones 4.7 y 4.8 se tiene que:

$$\dot{n}_{id}^1 = \gamma - \gamma \sum_{j=2}^E n_{id}^j - \gamma \sum_{j \neq 2} n_p^j - \gamma \sum_{j \neq 2} n_{tx}^j - (2\gamma + \lambda)n_{id}^1 + (\mu - \gamma)n_p^2 + (kv - \gamma)n_{tx}^2$$

Consideramos un nuevo vector de estados x sin incluir n_{id}^0 , es decir:

$$x = [n_{id}^1, n_p^1, n_{tx}^1, n_{id}^2, n_p^2, n_{tx}^2, \dots, n_{id}^E, n_p^E, n_{tx}^E]^T \in \mathbb{N}^{3 \times E}$$

Entonces el sistema definido en 4.4 con la restricción de densidad es equivalente al siguiente sistema lineal:

$$\dot{x} = Ax(t) + B, \quad x(t=0) = x_0. \quad (4.9)$$

Con matrices A y B de la forma (tomando el caso $E = 3$):

$$A = \begin{bmatrix} -2\gamma - \lambda & -\gamma & -\gamma & -\gamma & \mu - \gamma & kv - \gamma & -\gamma & -\gamma & -\gamma \\ \alpha\lambda & -\gamma - \mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ (1-\alpha)\lambda & 0 & -\gamma - kv & 0 & 0 & 0 & 0 & 0 & 0 \\ \gamma & 0 & 0 & -\gamma - \lambda & 0 & 0 & 0 & \mu & kv \\ 0 & \gamma & 0 & \alpha\lambda & -\gamma - \mu & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma & (1-\alpha)\lambda & 0 & -\gamma - kv & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma & 0 & \alpha\lambda & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma & (1-\alpha)\lambda & 0 & -kv \end{bmatrix}, B = \begin{bmatrix} \gamma \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.10)$$

4.3. Simulación del sistema

A los efectos de evaluar la aproximación fluida realizamos una simulación del sistema estocástico y la resolución del ODE determinístico con los parámetros detallados en la Tabla 4.4.

En las Figuras 4.1, 4.2 y 4.3, se muestra la evolución de la población de nodos de cada clase según el nivel de energía. La densidades de población simuladas aparecen como N_i^j/N y el modelo determinístico como n_i^j .

En todas las clases podemos ver que la aproximación fluida logra capturar tanto el componente transitorio como la respuesta en régimen de la dinámica.

Capítulo 4. Red IoT con Energy Harvesting

| Parámetro | Descripción | Valor | Unidad |
|-----------|---------------------------------|-------|----------|
| λ | Tasa de arribo de tareas | 1 | tareas/s |
| μ | Tasa de procesamiento de tareas | 0.5 | tareas/s |
| kv | Tasa de transmisión | 10.0 | tareas/s |
| E | Capacidad de batería | 3 | - |
| N | Cantidad de dispositivos IoT | 500 | - |
| T | Tiempo de simulación | 30 | s |

Tabla 4.4: Parámetros de la simulación

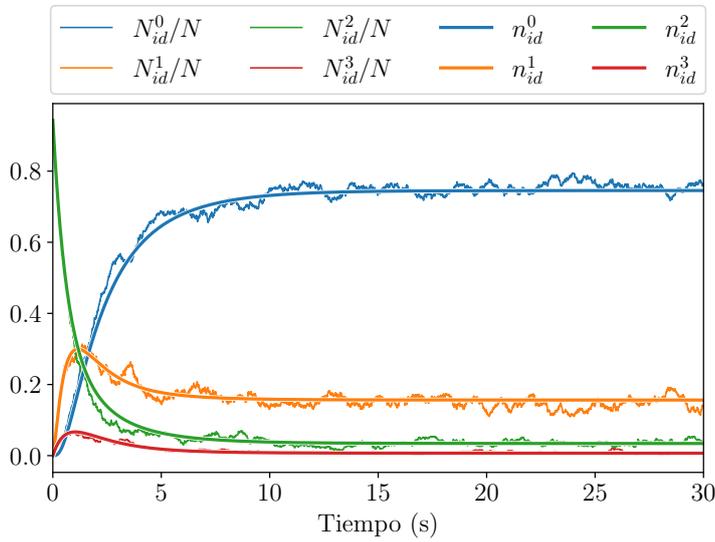


Figura 4.1: Evolución temporal de nodos en espera según nivel de carga.

4.4. Puntos de equilibrio

El punto de equilibrio del sistema (4.9) que llamaremos \tilde{x} , está dado por la siguiente expresión:

$$\tilde{x} = -A^{-1}B \quad (4.11)$$

En la Tabla 4.5 se muestran los resultados de una simulación del sistema estocástico y se compara con el punto de equilibrio del ODE calculado con la expresión 4.11. Se compara el punto de equilibrio del ODE con tres simulaciones estocásticas con los mismos parámetros pero variando el número total de dispositivos N . Para cada coordenada de la variable de estados x y el valor calculado de n_{id}^0 , se muestra: el valor de equilibrio del ODE así como la media \bar{n}_i^j y la desviación estándar $\sigma(\bar{n}_i^j)$ del sistema estocástico.

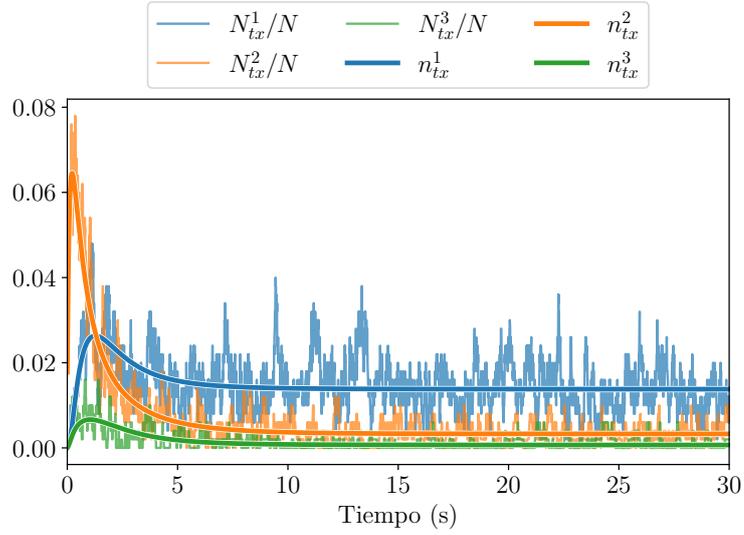


Figura 4.2: Evolución temporal de nodos transmitiendo según nivel de carga.

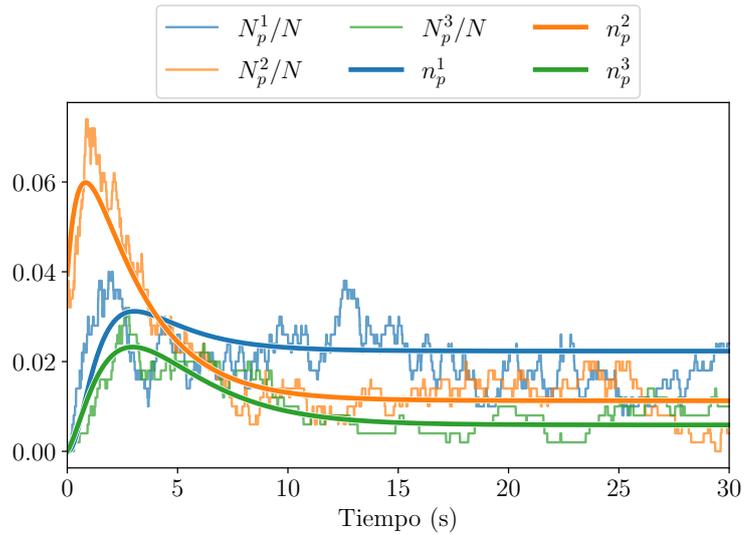


Figura 4.3: Evolución temporal de nodos procesando según nivel de carga.

A pesar de contar con un número finito N de dispositivos, se puede considerar que el límite fluido es una buena aproximación al comportamiento en régimen del sistema estocástico. Como era de esperar al aumentar N disminuye la varianza del proceso estocástico.

4.5. Estabilidad

Proposición 4.5.1. *La matriz F del sistema (4.4) no tiene valores propios con parte real estrictamente positiva, y tiene un valor propio $\lambda = 0$ simple.*

Capítulo 4. Red IoT con Energy Harvesting

| n_i^j | ODE | $N = 250$ | | $N = 500$ | | $N = 750$ | |
|------------|--------|---------------|-----------------|---------------|-----------------|---------------|-----------------|
| | | \bar{n}_i^j | $\sigma(n_i^j)$ | \bar{n}_i^j | $\sigma(n_i^j)$ | \bar{n}_i^j | $\sigma(n_i^j)$ |
| n_{id}^0 | 0.7455 | 0.7324 | 0.0279 | 0.7513 | 0.0161 | 0.7438 | 0.0137 |
| n_{id}^1 | 0.1635 | 0.1678 | 0.0248 | 0.1556 | 0.0147 | 0.1586 | 0.0131 |
| n_{id}^2 | 0.0557 | 0.0362 | 0.0122 | 0.0330 | 0.0077 | 0.0354 | 0.0057 |
| n_{id}^3 | 0.0011 | 0.0077 | 0.0058 | 0.0066 | 0.0033 | 0.0067 | 0.0026 |
| n_p^1 | 0.0225 | 0.0226 | 0.0086 | 0.0196 | 0.0063 | 0.0216 | 0.0056 |
| n_p^2 | 0.0148 | 0.0096 | 0.0059 | 0.0103 | 0.0047 | 0.0107 | 0.0032 |
| n_p^3 | 0.0073 | 0.0041 | 0.0039 | 0.0047 | 0.0030 | 0.0052 | 0.0024 |
| n_{tx}^1 | 0.0144 | 0.0148 | 0.0074 | 0.0147 | 0.0057 | 0.0138 | 0.0042 |
| n_{tx}^2 | 0.0052 | 0.0038 | 0.0040 | 0.0035 | 0.0027 | 0.0035 | 0.0022 |
| n_{tx}^3 | 0.0011 | 0.0010 | 0.0020 | 0.0007 | 0.0012 | 0.0007 | 0.0010 |

Tabla 4.5: Punto de equilibrio ODE vs Simulación Estocástica variando N

Demostración. $F = (f_{ij})$ es una matriz Metzler y además es irreducible dado que su grafo asociado está fuertemente conectado (ver Teorema 2.4.4).

Dado que el vector fila $\mathbb{1} \succ 0$ es vector propio de F asociado al valor propio $\lambda_0 = 0$, por el Teorema 2.4.5 tenemos que $\mu(F) = 0$ y λ_0 es simple. Dado que $\mu(F) = \max(\text{Re}(\lambda), \lambda \in \sigma(A)) = 0$ y el valor propio es simple, F no tiene valores propios con parte real estrictamente positiva. \square

Proposición 4.5.2. *La matriz A es Hurwitz.*

Demostración. La matriz $F \in \mathcal{M}_{n \times n}$ puede escribirse como sigue², definiendo submatrices de tamaños adecuados.

$$F = \begin{bmatrix} -\gamma & f_1 \\ \gamma & f_2 \\ \mathbf{0} & M \end{bmatrix} \quad (4.12)$$

En particular, $(f_1)_{1 \times (n-1)}$, $(f_2)_{1 \times (n-1)}$, $M_{(n-2) \times (n-1)}$, $\mathbf{0}_{(n-2) \times 1}$. Sabemos que $\mathbb{1}_n \cdot F = 0$, por lo que F tiene valor propio nulo³, con vector propio izquierdo asociado $\mathbb{1}_n$. Sea β un valor propio no nulo de F, con vector propio (derecho) asociado w . Un resultado de álgebra de matrices sobre la relación entre los vectores propios izquierdos y derechos, nos dice que

$$0 = \mathbb{1}_n \cdot w^T = \sum_i w_i \Rightarrow -w_1 = \sum_{i=2}^n w_i = \mathbb{1}_{n-1} \cdot w_{2-n}^T$$

²Por comodidad se utiliza n en lugar de $3 \times E + 1$

³Por la proposición 4.5.1 sabemos que este valor propio nulo es simple y el resto de los valores propios de F tienen parte real negativa estricta.

donde hemos definido el vector columna $w_{2-n}^T = [w_2, \dots, w_n]^T$. Por ser $w = [w_1, w_{2-n}]^T$ vector propio de F , se cumple que

$$F.w = \begin{bmatrix} -\gamma & f_1 \\ \gamma & f_2 \\ \mathbf{0} & M \end{bmatrix} .w = \begin{bmatrix} -\gamma.w_1 + f_1.w_{2-n} \\ \gamma.w_1 + f_2.w_{2-n} \\ M.w_{2-n} \end{bmatrix} = \beta \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_{3-n} \end{bmatrix} \quad (4.13)$$

con $w_{3-n} = [w_3, \dots, w_n]^T$.

En la misma línea que hicimos con la matriz F , observemos que podemos escribir la matriz $A_{(n-1) \times (n-1)}$ presentada en (4.10) como

$$A = \begin{bmatrix} f_2 \\ M \end{bmatrix} - \gamma \cdot \begin{bmatrix} 1 \\ \mathbf{0}_{(n-2) \times 1} \end{bmatrix} \cdot \mathbb{1}_{n-1} \quad (4.14)$$

Realicemos el siguiente cálculo:

$$\begin{aligned} A.w_{2-n} &= \begin{bmatrix} f_2 \\ M \end{bmatrix} .w_{2-n} - \gamma \cdot \begin{bmatrix} 1 \\ \mathbf{0}_{(n-2) \times 1} \end{bmatrix} \cdot \mathbb{1}_{n-1}^T .w_{2-n} \\ &= \begin{bmatrix} f_2.w_{2-n} \\ M.w_{2-n} \end{bmatrix} - \gamma \cdot \begin{bmatrix} 1 \\ \mathbf{0}_{(n-2) \times 1} \end{bmatrix} \cdot \mathbb{1}_{n-1}^T .w_{2-n} \end{aligned}$$

Usando (4.13), tenemos que

$$A.w_{2-n} = \begin{bmatrix} \beta.w_2 - \gamma.w_1 \\ \beta.w_{3-n} \end{bmatrix} - \gamma \cdot \begin{bmatrix} 1 \\ \mathbf{0}_{(n-2) \times 1} \end{bmatrix} \cdot (-w_1) = \beta \cdot \begin{bmatrix} w_2 \\ w_{3-n} \end{bmatrix} = \beta.w_{2-n}$$

Por lo que si β es valor propio no nulo de F , entonces también es valor propio de A . Además, tenemos que $F_{n \times n}$ tiene n valores propios contados con su multiplicidad y $A_{(n-1) \times (n-1)}$ tiene $n - 1$ valores propios. Como 0 es un valor propio simple de F , F tiene $n - 1$ valores propios con parte real negativa. A tiene entonces $n - 1$ valores propios con parte real negativa por lo que A es Hurwitz. \square

En la Figura 4.4 se muestran los valores propios de las matrices F y A del sistema especificado en Tabla 4.4, donde verificamos que los valores propios de A son los valores propios de F sin contar el valor propio nulo, y además todos tienen parte real negativa. En la Figura 4.5 se muestran los valores propios aumentando la capacidad de la batería con $E = 24$.

Al ser A Hurwitz, el sistema (4.9) tiene un único punto de equilibrio y es asintóticamente estable. Además, A es invertible puesto que no tiene valor propio nulo y $\tilde{x} = -A^{-1}B$ queda bien definido.

Dado que F es Metzler, aplicando el Teorema 2.4.3 tenemos que el sistema definido en (4.4) es positivo.

Capítulo 4. Red IoT con Energy Harvesting

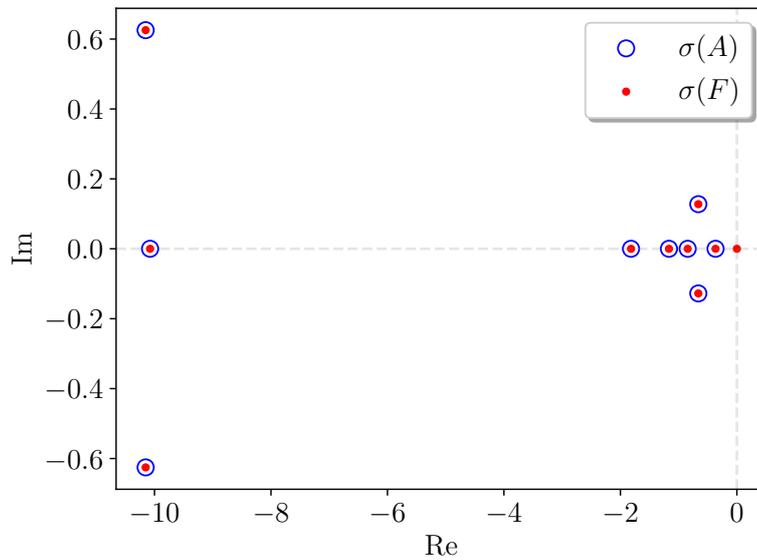


Figura 4.4: Ubicación de los valores propios de F y A para el sistema especificado en Tabla 4.4.

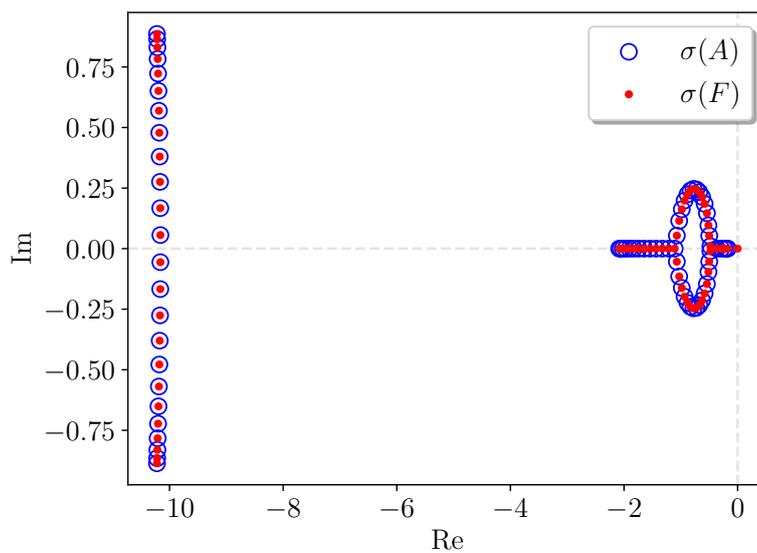


Figura 4.5: Ubicación de los valores propios de F y A con $E = 24$, y con el resto de los parámetros especificados en Tabla 4.4.

4.6. Calidad de servicio

Una vez verificado que la aproximación fluida es razonable respecto al sistema estocástico nos interesa poder estimar ciertas métricas de calidad de servicio de la red IoT. En el capítulo 3 se introdujeron las pérdidas por ocupación. Este es el caso cuando llegan tareas a nodos ocupados (procesando o transmitiendo). En esta sección incorporamos el concepto de pérdidas por energía, que son las pérdidas que

ocurren cuando arriban tareas a nodos en espera sin energía y que por lo tanto no pueden ni transmitirse ni procesarse localmente.

En primer lugar vamos a ver que a partir del vector de estados x es posible construir distintas salidas del sistema que serán utilizadas más adelante.

4.6.1. Salidas del sistema

A partir del vector de estados se pueden construir distintas salidas de utilidad. En particular tenemos que:

- $n_p = [0, 1, 0, 0, 1, 0, \dots, 0, 1, 0]x(t)$.
- $n_{tx} = [0, 0, 1, 1, 0, 0, 1, \dots, 0, 0, 1]x(t)$.
- $n_{id} = 1 - [0, 1, 1, 0, 1, 1, \dots, 0, 1, 1]x(t)$.
- $n_{id}^0 = 1 - \mathbb{1}x(t)$

Definiremos la carga de energía del sistema (SoC por “State of Charge”) como la energía total disponible sobre la capacidad de energía soportada.

$$\text{SoC}(t) \triangleq \frac{\sum_{m=1}^N E_m(t)}{E \times N} \quad (4.15)$$

La cantidad total de unidades de energía en el sistema estocástico la podemos aproximar con las densidades de los distintos niveles de energía, entonces la energía total del sistema la podemos calcular como:

$$\sum_{m=1}^N E_m(t) = \sum_{j=1}^E j(n_p^j + n_{tx}^j + n_{id}^j)N = N[1, 1, 1, 2, 2, 2, 3, 3, 3, \dots, E, E, E]x(t)$$

A partir de la definición dada de SoC en 4.15 tenemos que:

$$\text{SoC}(t) = \left[\frac{1}{E}, \frac{1}{E}, \frac{1}{E}, \frac{2}{E}, \frac{2}{E}, \frac{2}{E}, \dots, \frac{E}{E}, \frac{E}{E}, \frac{E}{E} \right]^T x(t)$$

4.6.2. Pérdidas por energía

Las tareas que se pierden por energía son aquellas que arriban a un nodo en espera con batería descargada. Supongamos que en determinado momento hay N_{id}^0 nodos en espera sin carga. Una nueva pérdida por energía se dará cuando arribe una tarea a cualquiera de esos nodos. Por lo tanto el tiempo entre pérdidas es una variable aleatoria con distribución exponencial con tasa λN_{id}^0 . La tasa de pérdidas por energía y por dispositivo es $l_e(t) = \lambda n_{id}^0(t)$ que se puede escribir a partir de la variable de estados como:

$$l_e(t) = \lambda(1 - \mathbb{1}x(t)) \quad (4.16)$$

Capítulo 4. Red IoT con Energy Harvesting

| Parámetro | Descripción | Escenario I | Escenario II |
|-----------------|--|-------------------------|-------------------------|
| λ | Tasa arribo tareas | 0.3 [s ⁻¹] | 1.0 [s ⁻¹] |
| γ | Tasa arribo energía | 0.1 [s ⁻¹] | 1.4 [s ⁻¹] |
| μ | Tasa de procesamiento | 0.5 [s ⁻¹] | 0.5 [s ⁻¹] |
| kv | Tasa de transmisión | 10.0 [s ⁻¹] | 10.0 [s ⁻¹] |
| $n_{id}^0(t_0)$ | Densidad nodos en espera descargados t_0 | 0 | 1 |
| $n_{id}^E(t_0)$ | Densidad nodos en espera cargados t_0 | 1 | 0 |
| E | Niveles de batería | 15 | 15 |
| N | Cantidad de dispositivos IoT | 500 | 500 |
| T | Tiempo de simulación | 300 [s] | 100 [s] |

Tabla 4.6: Parámetros de simulaciones en escenarios de carga y descarga.

4.6.3. Pérdidas por ocupación

La pérdida por ocupación se produce cuando arriba una tarea a un nodo que se encuentra procesando o transmitiendo. El tiempo entre pérdidas por ocupación es una variable aleatoria con distribución exponencial con tasa $\lambda(N_p + N_{tx})$, que surge de considerar el mínimo de N_p variables aleatorias exponenciales de tasa λ y N_{tx} exponenciales de tasa λ . La tasa de pérdidas por ocupación y por dispositivo es $l_o(t) = \lambda(n_p(t) + n_{tx}(t))$ que se puede escribir a partir de la variable de estados como:

$$l_o(t) = \lambda \left(\sum_{j=1}^E e_p^j + \sum_{j=1}^E e_{tx}^j \right) x(t) \quad (4.17)$$

4.7. Casos de ejemplo

Nos interesa ahora estudiar la evolución de las variables agregadas, es decir $n_{tx}(t)$ y $n_p(t)$ así como $SoC(t)$. Pasar de un estado de $3 \times E$ dimensiones a otro de 3 dimensiones nos permitirá visualizar gráficamente la evolución del sistema así como determinar algunos aspectos de calidad de servicio.

Analizaremos en particular dos escenarios distintos. En la sección 4.7.1 analizamos el caso donde la generación de energía es baja con respecto a la demanda de procesamiento de tareas. Posteriormente, en la sección 4.7.2, veremos el caso contrario donde la generación de energía es muy alta respecto a la demanda.

En ambos escenarios veremos cómo se comporta la aproximación fluida respecto a la simulación estocástica y también analizaremos la evolución de las métricas de QoS a lo largo del tiempo.

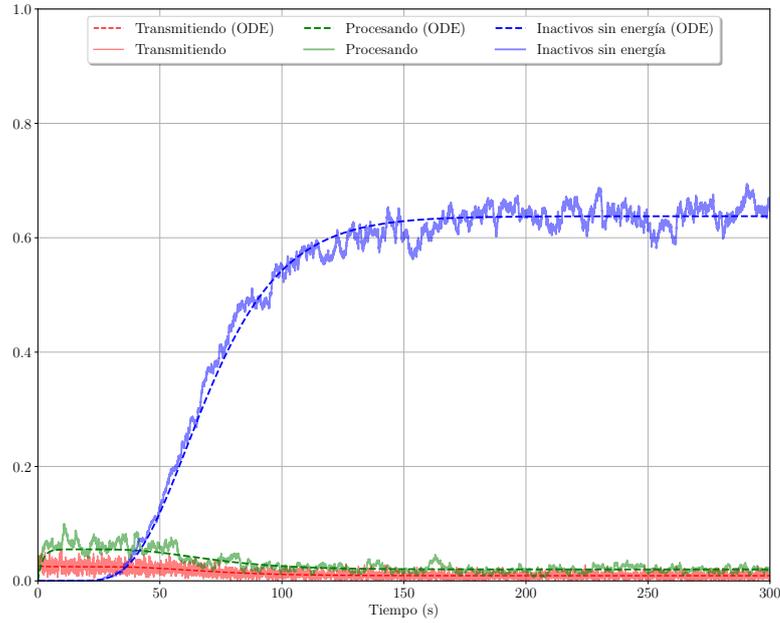


Figura 4.6: Evolución temporal de nodos en espera sin energía, nodos procesando y transmitiendo.

4.7.1. Escenario I: descarga de red IoT

Consideramos el caso de una red IoT con los parámetros del Escenario I que aparecen en la Tabla 4.6. Inicialmente todos los nodos se encuentran en espera y completamente cargados, es decir $n_{id}^E(t_0) = 1$. Tomamos un valor de γ relativamente chico respecto a λ de forma de descargar rápidamente el sistema, con $\gamma = \lambda/3$.

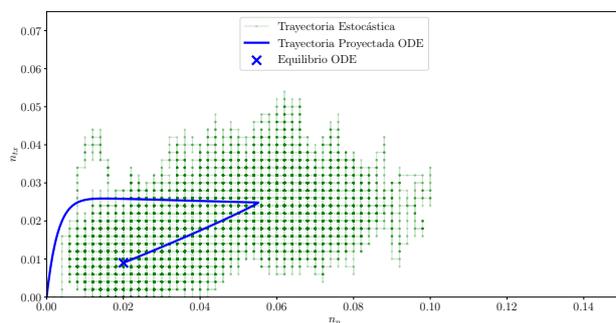
Los valores esperados de las distintas variables agregadas se pueden obtener a partir del punto de equilibrio calculado con la expresión (4.11). Los valores en equilibrio son: $\tilde{n}_p = 0.019$, $\tilde{n}_{tx} = 0.009$ y $\tilde{n}_{id} = 0.971$. Tenemos también que $\tilde{n}_{id}^0 = 0.64$ y $\tilde{SoC} = 0.039$.

En la Figura 4.6 se muestra la evolución temporal de las poblaciones agregadas n_p, n_{tx} así como de n_{id}^0 tanto para la simulación estocástica como para el ODE. Inicialmente la población de nodos en espera descargados es nula, luego cerca de los 25 s de simulación esta población comienza a aumentar hasta convertirse finalmente en la predominante con más del 60% de la población total. En la Figura 4.7a representamos la evolución de las variables agregadas n_p y n_{tx} .

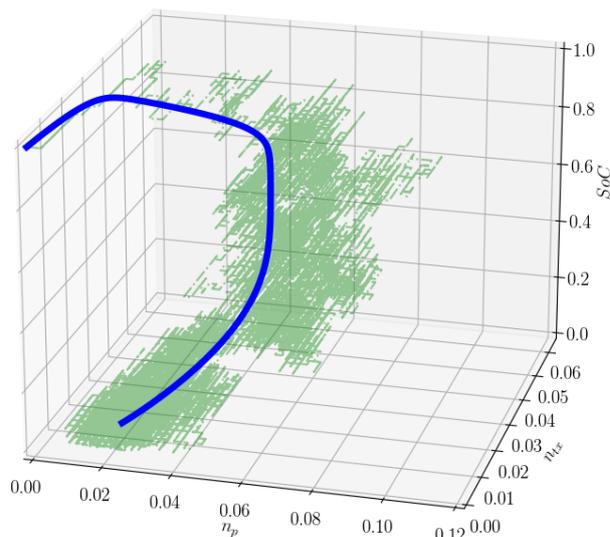
En la Figura 4.7b agregamos una dimensión de energía a través del estado de carga SoC donde se puede apreciar que el sistema parte de un estado completamente cargado, hay una primera etapa donde la afectación de energía no es importante y luego que comienzan a surgir nodos descargados el punto de operación (n_p, n_{tx}) cambia sustancialmente debido a que nacen menos tareas por falta de energía.

El estado de carga del sistema que se muestra en la Figura 4.8 es correctamente representado por el modelo, tanto en su dinámica como en su punto de equilibrio.

Capítulo 4. Red IoT con Energy Harvesting



(a) Diagrama (n_p, n_{tx})



(b) Diagrama (n_p, n_{tx}, SoC), en azul vemos la trayectoria del ODE de las variables agregadas y en verde la simulación estocástica.

Figura 4.7: Evolución de las variables agregadas en escenario de descarga.

Las pérdidas por energía y ocupación, tanto para la simulación estocástica y la aproximación fluida se muestran en la Figura 4.9. A los efectos de calcular la tasa de pérdidas en la simulación estocástica se contaron los eventos de pérdida que sucedieron en una ventana de 1 s. Como era de esperar, inicialmente no hay pérdidas por energía porque los nodos están cargados y luego cuando el sistema se descarga las tareas perdidas por energía superan el 60%. Por el contrario las pérdidas por ocupación pasan aproximadamente del 8% al 4%, la falta de energía hace que nazcan menos nodos transmisores y procesadores, y por consiguiente las pérdidas por ocupación disminuyen.

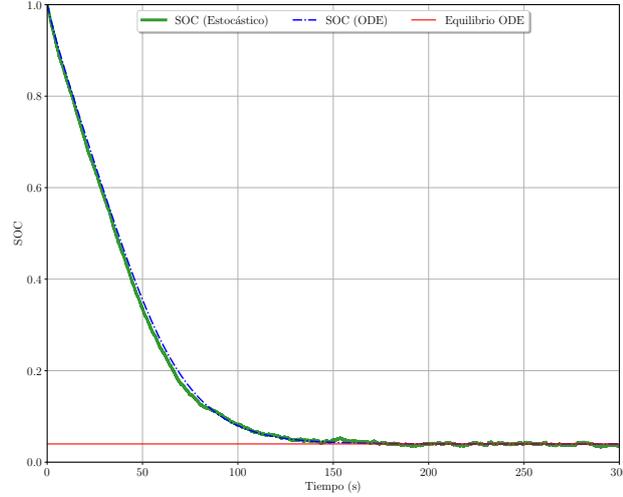


Figura 4.8: SoC de red IoT en escenario de descarga.

4.7.2. Escenario II: carga de red IoT

A los efectos de verificar el modelo para situaciones de carga de energía, en este ejemplo simulamos una red con parámetros similares al experimento anterior (ver Tabla 4.6) pero con mayor *harvesting* de energía, con tasa $\gamma = 1.4\lambda$. Inicialmente todos los nodos se encuentran completamente descargados y en espera.

Esta vez en lugar de visualizar la evolución de las clases agregadas representamos para cada clase la población según el nivel de carga. En las Figuras 4.10, 4.11 y 4.12, las densidades de población de la simulación estocástica aparecen como N_i^j/N y las aproximaciones fluidas como n_i^j .

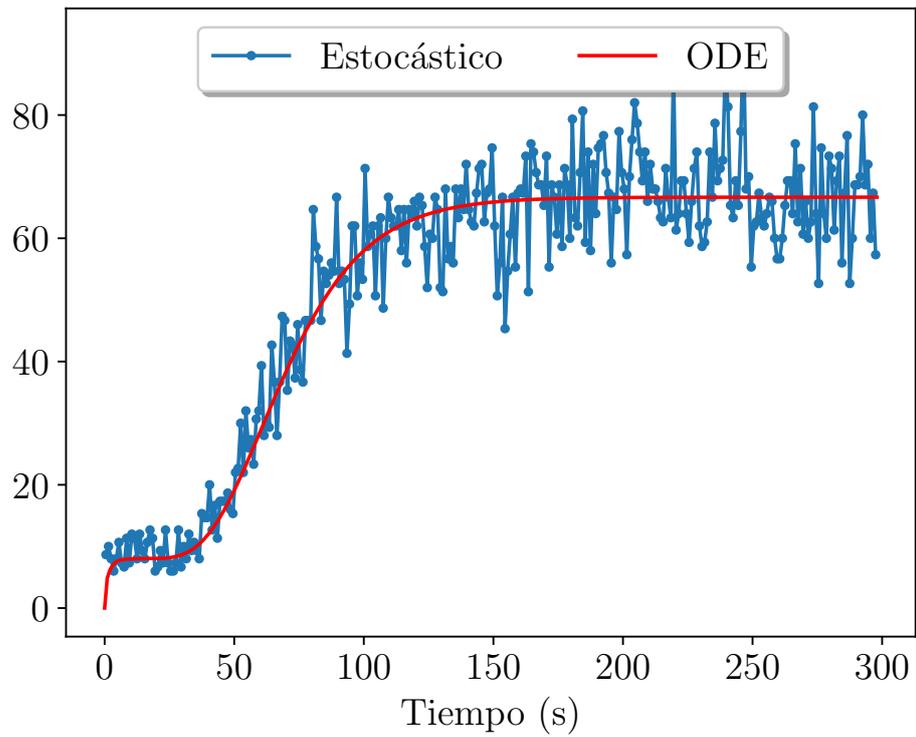
Inicialmente toda la población se compone de nodos en espera descargados y luego paulatinamente con la llegada de unidades de energía aparecen nodos con mayores niveles de carga y distinta clase. Dado que para llegar al nivel de energía j el nodo previamente debe estar en el nivel $j - 1$ las curvas de densidad según nivel de carga aparecen desfasadas entre ellas. La aproximación fluida se muestra razonable tanto en el componente transitorio como el valor en régimen para cada una de las coordenadas del vector de estados, aún con una cantidad limitada de dispositivos ($N = 500$).

En las Figuras 4.14a y 4.14b se puede ver la evolución de las variables agregadas en el escenario de carga, y el punto de equilibrio de las variables agregadas (n_p, n_{tx}, SoC) en la Tabla 4.7.

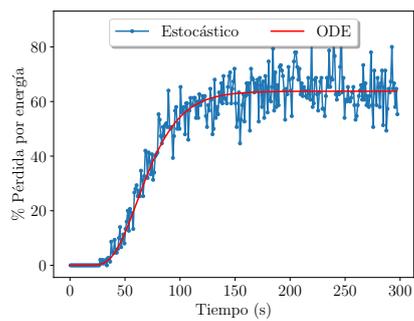
El flujo de energía que llega al sistema en media es $\lambda N = 700$, que supera el flujo de energía consumido $(\mu n_p + kv n_{tx})N \simeq 94.52$. La diferencia de flujo no puede aprovecharse y se descarta, y vale

$$(\lambda - \mu n_p + kv n_{tx})N \simeq 605.48.$$

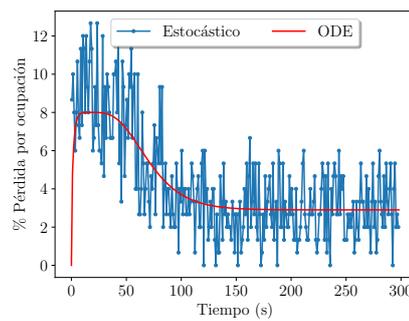
El descarte de energía también se puede obtener de las coordenadas de equilibrio n_i^E a partir de la expresión $\gamma(n_{id}^E + n_{pl}^E + n_{tx}^E)N \simeq 605.48$. Todas las unidades



(a)



(b)



(c)

Figura 4.9: Porcentajes de pérdidas en escenario de descarga (a) totales (b) por energía (c) por ocupación.

4.7. Casos de ejemplo

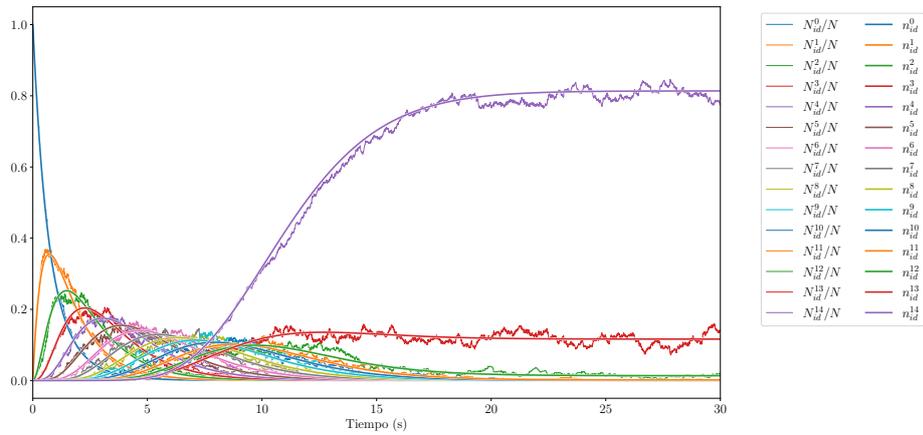


Figura 4.10: Densidad de nodos en espera según nivel de carga.

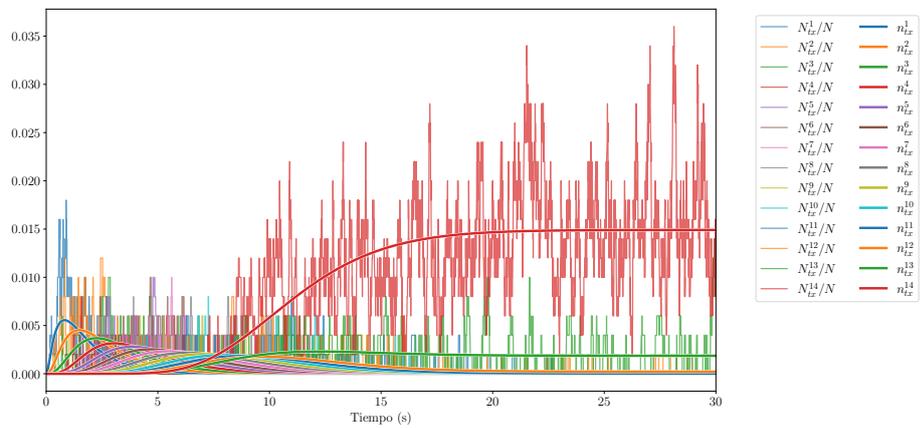


Figura 4.11: Densidad de nodos transmitiendo según nivel de carga.

| Variable | Valor en equilibrio |
|----------|---------------------|
| n_p | 0.0378 |
| n_{tx} | 0.0170 |
| SoC | 0.9890 |

Tabla 4.7: Valores de equilibrio de variables agregadas del experimento de carga.

de energía que lleguen a nodos con nivel de energía E se descartan, la deducción se hace considerando los tiempos exponenciales del arribo de energía sobre esos nodos.

El SoC calculado a partir del vector de estados de equilibrio vale 0.98909 (ver Figura 4.15) y las pérdidas por concepto de energía se mantienen muy bajas (ver Figura 4.16).

Capítulo 4. Red IoT con Energy Harvesting

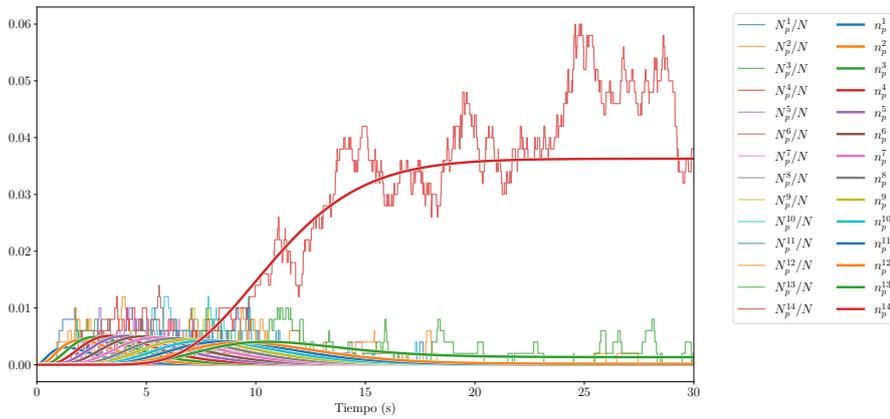


Figura 4.12: Densidad de nodos procesando según nivel de carga.

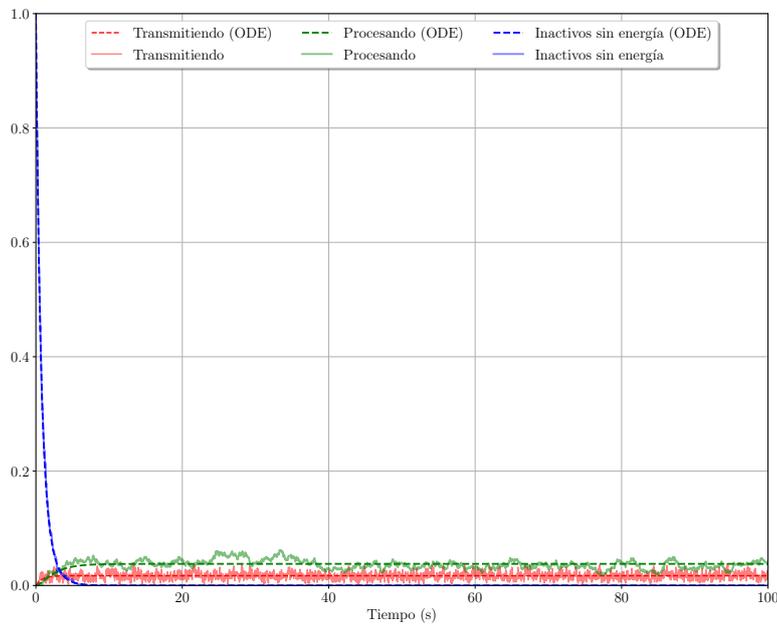


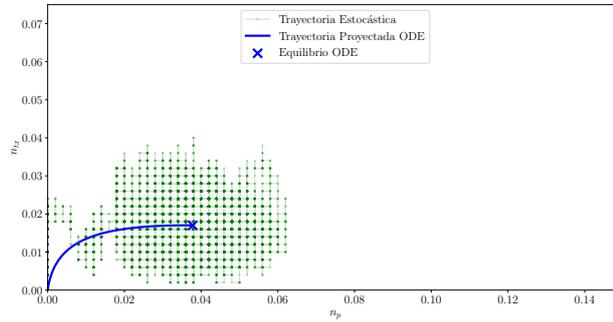
Figura 4.13: Evolución temporal de nodos en espera sin energía, nodos procesando y transmitiendo.

4.8. Conclusiones

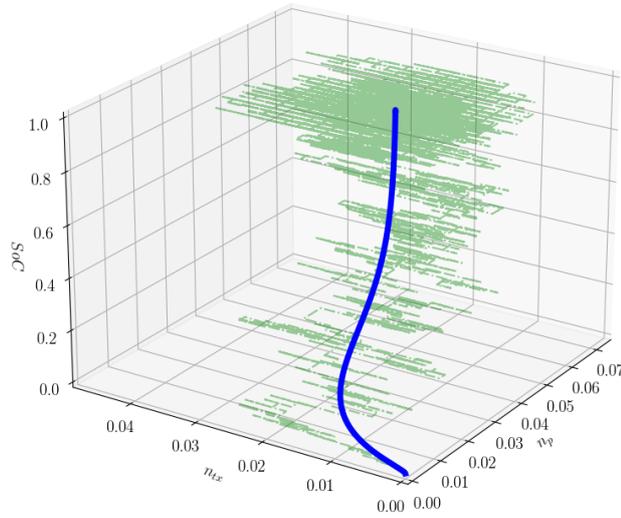
En este capítulo hemos presentado un modelo de energía basado en un proceso de Markov en tiempo continuo, considerando poblaciones de nodos según niveles de carga e intensidades de transición entre clases y niveles de energía.

Aplicando el método de la aproximación fluida al proceso de Markov se logra obtener un modelo determinístico que surge de resolver un sistema de ecuaciones diferenciales ordinarias.

A partir de ciertas propiedades de la matriz de intensidades (matrices Metzler y



(a) Diagrama (n_p, n_{tx})



(b) Diagrama (n_p, n_{tx}, SoC) , en azul aparece la trayectoria del ODE de las variables agregadas y en verde la simulación estocástica.

Figura 4.14: Evolución de las variables agregadas en escenario de carga.

suma cero de las columnas) se logró demostrar la estabilidad asintótica del punto de equilibrio. Además se verificó que el sistema es positivo y el simplex con las dimensiones del vector de estados es un conjunto invariante.

Los resultados comparados entre simulaciones estocásticas y el modelo determinístico muestran que la aproximación fluida es razonable tanto para el componente transitorio como para el valor en régimen.

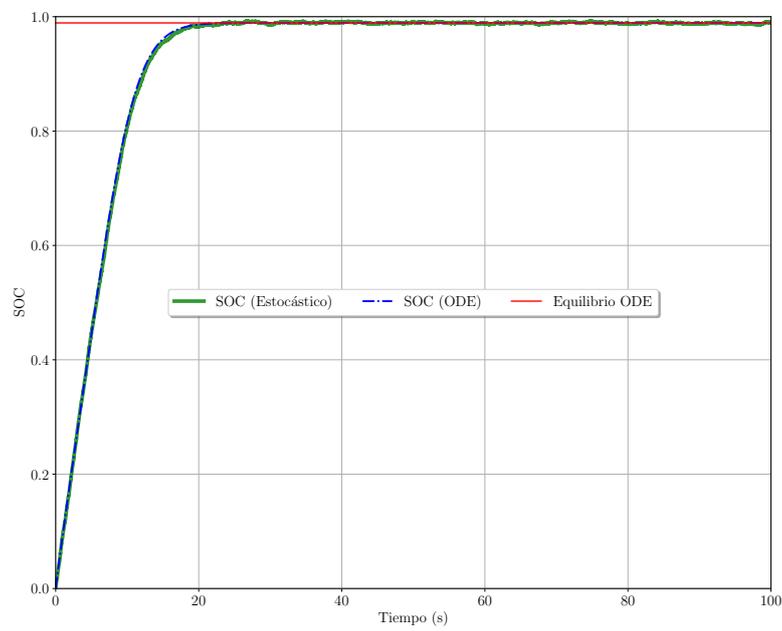
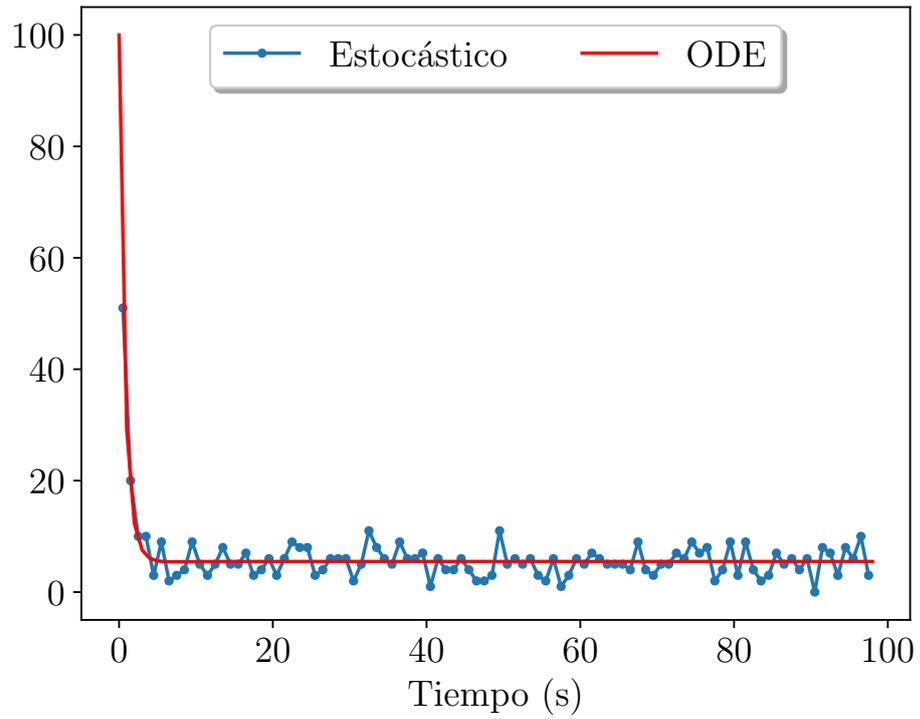
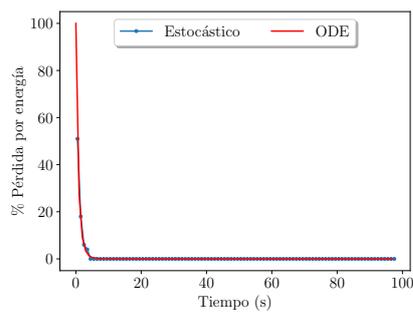


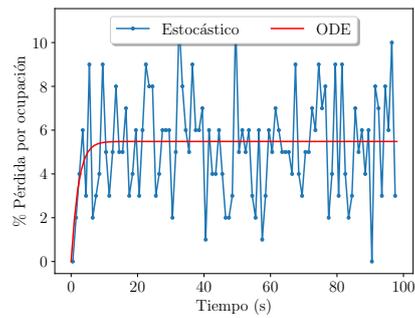
Figura 4.15: $SoC(t)$



(a)



(b)



(c)

Figura 4.16: Porcentaje de pérdidas en escenario de carga (a) totales (b) por ocupación (c) por energía.

Esta página ha sido intencionalmente dejada en blanco.

Conclusiones

El *Edge Computing* combinado con el *Cloud Computing*, se apresta a transformar radicalmente la calidad de servicio de aplicaciones que requieren baja latencia y procesamiento intensivo. Entre los casos de uso beneficiados se encuentran la Realidad Virtual, IoT, Vehículos Conectados, Telemedicina, entre otros.

Las redes IoT que requieran procesamiento intensivo para viabilizar este tipo de aplicaciones además de beneficiarse de una gran disponibilidad de unidades de procesamiento en el Cloud, pueden verse beneficiadas por el Edge para reducir la latencia de las comunicaciones y por consiguiente el tiempo total de ejecución de tareas.

El asunto de cómo distribuir la demanda de procesamiento entre las distintas capas Local-Edge-Cloud es uno de los problemas planteados actualmente. Uno de los principales desafíos a la hora de resolver estos problemas es la posibilidad de escalar con la cantidad de dispositivos (EDs y ECs) así como el tiempo de resolución. Cuando un sistema de decisión trabaje en línea para resolver cómo distribuir el procesamiento de las tareas, el tiempo de offloading no solo incluirá los tiempos de ejecución y transmisión sino también el tiempo de decisión del algoritmo de offloading, por lo que es vital reducir estos tiempos.

En esta tesis se investigó un enfoque hasta ahora no explorado en el Estado del Arte, con la formulación de un proceso de Markov escalado con la cantidad de usuarios de la red IoT. La aproximación fluida resulta en la solución de un ODE determinístico que pudo caracterizarse con técnicas clásicas. El proceso escalado converge en probabilidad cuando N tiende a infinito a la solución de un ODE determinístico. Aún cuando no se disponga de una cantidad infinita de dispositivo se logró verificar que un número grande pero finito de dispositivos es una aproximación razonable al proceso estocástico.

Aquí es importante hacer algunas consideraciones sobre modelado en general. Un modelo busca representar cierto fenómeno con el objetivo de poder comprenderlo e incluso actuar sobre él para algún fin. Los modelos parten de ciertas hipótesis sobre el funcionamiento de un sistema de forma de simplificar el análisis y obtener alguna conclusión sobre alguna pregunta que nos podamos realizar. Naturalmente se pueden formular modelos más complejos que puedan representar más fielmente la naturaleza del fenómeno, pero esto puede ser a costa de complejizar el análisis y tener que recurrir a por ejemplo simulaciones, técnicas heurísticas u otro tipo de técnicas difícilmente interpretables. La complejidad aumenta si se deben definir tecnologías o protocolos específicos, y más si los dispositivos son distintos entre sí. Además las conclusiones pueden variar si se utiliza una u otra tecnología. Por esta

Capítulo 4. Red IoT con Energy Harvesting

razón decidimos trabajar con el modelo más sencillo posible que pudiera capturar el fenómeno, con hipótesis razonables, y al mismo tiempo obtener expresiones analíticas para lograr optimizar el funcionamiento del sistema en su conjunto.

Entre las características que se buscaron representar en el modelado podemos destacar: i) la naturaleza estocástica de la ocurrencia de distintos tipos de eventos (comienzo y fin de procesamiento o transmisión, llegada de tareas, etc.), ii) el impacto de compartir el medio físico entre distintos dispositivos IoT sobre el throughput alcanzado en la transmisión, en particular se buscó capturar la saturación del canal a medida que aumenta la cantidad de transmisores por el aumento de colisiones y la adaptación del tiempo de backoff en protocolos como ALOHA o CSMA, iii) un factor de offloading que pueda ser utilizado como parámetro de control de la red, iv) la posibilidad de analizar redes IoT arbitrariamente grandes en tiempos acotados.

El proceso estocástico se analiza a través de la aproximación fluida del proceso de Markov escalado con la cantidad de dispositivos. Esto requirió algunas hipótesis como el suponer la homogeneidad de los parámetros entre todos los dispositivos de la red ($\lambda, \mu, c, k, \alpha, \text{etc.}$). En una línea de trabajo futuro se podría simular el efecto de tener valores de parámetros distintos entre los elementos de la red, y estudiar si por ejemplo considerar un valor medio de estos parámetros permite seguir representando razonablemente el comportamiento del sistema.

El modelo de throughput utilizado dividió el denominado triángulo factible en dos zonas según el estado del canal. En la zona inferior del triángulo el throughput por usuario se mantiene constante mientras que en la parte superior del triángulo, el canal se encuentra saturado y el throughput por usuario disminuye a medida que se incorporan nuevos transmisores. Cada zona alberga entonces una dinámica particular, y las trayectorias se analizaron considerando ambas dinámicas en un esquema de sistema conmutado.

Como primer paso, fue preciso determinar la ubicación de los P.E. y la estabilidad de cada subsistema. Un análisis del sistema conmutado permitió determinar que para todo parámetro y cualquier condición inicial dentro del triángulo factible, las trayectorias convergen a un único punto de equilibrio, ya sea n_{sat} o n_{lin} .

Todas las trayectorias se mantienen dentro del triángulo, es decir que el triángulo es invariante frente a la dinámica, y las poblaciones de nodos transmitiendo, procesando e inactivos forman una densidad.

Se obtuvo una expresión analítica del P.E. de cada subsistema así como condiciones de estabilidad para los mismos. El P.E. del sistema I es asintóticamente estable. En cambio, según la combinación de parámetros, el P.E. del sistema II puede ser estable o inestable.

Un aspecto muy interesante de los subsistemas es el estudio del lugar geométrico del P.E. según el factor de offloading α . Este lugar geométrico para cada subsistema se pudo obtener representando los puntos de equilibrio en el plano complejo e identificando una transformación de Möbius.

El punto de equilibrio del sistema conmutado surge de analizar cuidadosamente la ubicación de los puntos de equilibrio de cada subsistema en conjunto con la estabilidad del sistema II. En general, el punto de equilibrio del sistema conmutado

corresponde al P.E. del sistema I, pero con cierta combinación de parámetros el P.E. corresponde al P.E. del sistema II.

La transformación $\Gamma(\alpha)$ que lleva el factor de offloading al P.E. es una biyección, si se busca que el punto de operación de la red esté en algún punto (y_0, y_1) en \mathbf{C}^{lin} o \mathbf{C}^{sat} puedo estar seguro que existe un único α^* tal que $\Gamma(\alpha^*) = (y_0, y_1)$.

Una vez comprendida la dinámica del sistema conmutado el objetivo es poder determinar el mejor punto de operación de la red, según alguna métrica de calidad de servicio.

A esta altura es posible pensar distintas funciones de costo para optimizar según el factor de offloading. La idea general es construir una función de costo convexa, dependiente de n_{tx} y n_p . De esta manera se puede formular un problema de optimización convexa con restricciones de calidad de servicio.

Se formularon dos métricas de QoS: tasa de pérdida y tiempo de retardo, que dependen de las coordenadas n_{tx} y n_p . El conjunto de restricciones de QoS expresadas en el plano $n_{tx} - n_p$ forma un conjunto convexo ya que resulta de intersectar semiplanos. Además, se agregó como restricción el conjunto de los P.E. \mathbf{C}^{lin} o \mathbf{C}^{sat} , de forma que el punto de operación fuera alcanzable para algún $\alpha \in [0, 1]$.

La función de costo elegida, como prueba de concepto, representa el costo de procesar en el Fog y el Cloud. Se asume que procesar localmente no conlleva un costo, y luego, procesar en Fog o en el Cloud tiene un determinado precio (unidades monetarias / tareas. s^{-1}). Las tareas transmitidas son procesadas en el Fog hasta que se completa la capacidad de procesamiento en este nivel y lo que sigue se procesa en el Cloud. De esta manera, el costo solo depende de n_{tx} , y si n_{tx} aumenta también lo hará el costo de operación de la red.

Para verificar la aproximación fluida se diseñó e implementó un simulador de eventos con la dinámica del modelo. Se pudo verificar, con un número finito de dispositivos ($N = 500$) que: i) se logra captar el transitorio, ii) se logra captar el punto de equilibrio, iii) la evolución de la calidad de servicio,

Respecto al procedimiento de optimización, una simulación del proceso estocástico con el factor α obtenido permitió verificar que el punto de operación era el buscado cumpliendo además con las restricciones de QoS.

Con lo anterior se pudo caracterizar y encontrar un procedimiento para hallar un factor de offloading óptimo para cierta función convexa contemplando restricciones de calidad de servicio.

En la segunda parte de la tesis, se analiza el caso de una red IoT con *energy harvesting*. Cada nodo cuenta con cierta capacidad de almacenamiento y puede cosechar energía del entorno a cierta tasa. A su vez, transmitir o procesar consume recursos de la batería. Si no hay energía no pueden procesarse o transmitirse las tareas. La energía que llega cuando la batería se encuentra completamente cargada se descarta.

Se formula un proceso de Markov para estudiar la población de nodos según: estado y nivel de energía. Se divide la población en $3 \times E + 1$ clases donde E es la capacidad de almacenamiento. Luego de calcular la matriz de intensidades del proceso de Markov escalado se realiza la aproximación de límite fluido para

Capítulo 4. Red IoT con Energy Harvesting

obtener una aproximación determinística solución de un ODE.

Debido a la cantidad elevada de dimensiones que son difíciles de visualizar, se consideran las variables agregadas n_{tx} , n_p y SoC . Donde SoC (por State of Charge) es la cantidad de energía de la red IoT sobre la capacidad de almacenamiento total.

Se logró demostrar la existencia de un único punto de equilibrio del ODE determinístico así como su estabilidad asintótica a través de resultados de sistemas positivos y matrices Metzler.

Se analizaron dos escenarios variando γ , la tasa de generación de energía, para comparar la simulación estocástica contra la aproximación determinística. Los resultados obtenidos permiten afirmar que el límite fluido es una muy buena aproximación para obtener el punto de equilibrio así como la evolución de las coordenadas del vector de estados.

Se analizaron también métricas de calidad de servicio como son pérdidas por ocupación y pérdidas por energía. Las simulaciones realizadas nos permite afirmar que la aproximación fluida logra capturar el transitorio y el comportamiento en régimen de éstas pérdidas.

Dentro del alcance de esta segunda parte, no quedó incluido el modelo de canal compartido en zona de saturación ni la formulación de un problema de optimización para encontrar un factor de offloading óptimo, por lo que estos elementos pueden ser una línea de trabajo futuro.

A lo largo de este trabajo se analizaron dos casos de estudio referidos a una red IoT que puede realizar procesamiento de tareas en tres capas: Local-Fog-Cloud. Se modeló el proceso estocástico como un proceso de Markov que escala con la cantidad de dispositivos IoT. En el primer caso de estudio el modelo toma en cuenta la saturación del canal, en el segundo caso se estudia un esquema de energy harvesting.

En ambos casos se discute acerca del punto de operación de la red según los parámetros del sistema, en particular el punto de equilibrio y su estabilidad. Las simulaciones realizadas del proceso estocástica permitieron validar la aproximación fluida para el análisis de redes IoT con posibilidades de computar en distintas capas de procesamiento.

Apéndice A

Proposiciones

Proposición A.0.1. Sea $f : \mathbb{C}_\infty \rightarrow \mathbb{C}_\infty$ una transformación de Möbius de la forma $f(z) = \frac{az+b}{cz+d}$, con $a, b, c, d \in \mathbb{C}$, $ad - bc \neq 0$. Si $c \neq 0$ definimos $f(\infty) = \frac{a}{c}$ y $f(-d/c) = \infty$. Sea L el conjunto de rectas euclídeas en el plano complejo, si $\text{Im}(c) = 0$ y $\text{Im}(d) = 0$ entonces f mapea $\mathbb{R} \cup \{\infty\}$ en una recta en $L \cup \{\infty\}$.

Demostración. Una transformación de Möbius puede definirse completamente con el mapeo de tres puntos. Sean $w_0, w_1, w_2 \in \mathbb{C}_\infty$ puntos arbitrarios y sus mapeos $f_0 = f(w_0)$, $f_1 = f(w_1)$, $f_2 = f(w_2)$. $f(\cdot)$ mapeará la recta real en una recta del plano complejo si para cualquier terna $w_0, w_1, w_2 \in \mathbb{R} \cup \{\infty\}$ se tiene que $\frac{f_2 - f_0}{f_2 - f_1} \in \mathbb{R}$, esto es si $\overrightarrow{f_0 f_2}$ y $\overrightarrow{f_1 f_2}$ son colineales. Eligiendo $w_0 = 0 + 0j$, $w_1 = 1 + 0j$ y $w_2 = \infty$ entonces $f_0 = b/d$, $f_1 = \frac{a+b}{c+d}$ y $f_2 = a/c$, por lo tanto:

$$\frac{f_2 - f_0}{f_2 - f_1} = \frac{c + d}{d}$$

Finalmente, si $\text{Im}(c) = 0$ y $\text{Im}(d) = 0$ entonces $\frac{f_2 - f_0}{f_2 - f_1} \in \mathbb{R}$. □

Proposición A.0.2. El lugar geométrico de los puntos de equilibrio del Sistema II presentará una discontinuidad para algún α con $0 \leq \alpha \leq 1$ si $k_1 v \leq \lambda$.

Demostración. El lugar geométrico presentará una discontinuidad para algún α con $0 \leq \alpha \leq 1$ si $0 \leq \frac{\mu(\lambda - k_1 v)}{\lambda(k_1 v + \mu)} \leq 1$, esto ocurre exactamente cuando se cumple que:

$$k_1 v \leq \lambda \tag{A.1}$$

$$\frac{\mu(\lambda - k_1 v)}{\lambda(k_1 v + \mu)} \leq 1 \tag{A.2}$$

La segunda condición se puede ver que se verifica siempre (considerando que los parámetros del sistema k_1, v, λ y μ son positivos):

$$\begin{aligned} \frac{\mu(\lambda - k_1 v)}{\lambda(k_1 v + \mu)} \leq 1 &\iff \mu(\lambda - k_1 v) \leq \lambda(k_1 v + \mu) \\ &\iff -\mu k_1 v \leq \lambda k_1 v \end{aligned}$$

□

Apéndice A. Proposiciones

Proposición A.0.3. *Los valores propios γ_1 y γ_2 de la matriz fundamental del Sistema I son reales $\forall \alpha \in [0, 1]$.*

Demostración. El polinomio característico de de la matriz fundamental A está dado por $\gamma^2 - tr(A)\gamma + det(A)$. Las raíces del polinomio son reales si $tr(A)^2 \geq 4det(A)$.

$$\begin{aligned} tr(A)^2 &\geq 4det(A) \\ \iff \mu^2 + \lambda^2 + k^2v^2 + 2\mu\lambda + 2\mu kv + 2\lambda kv &\geq 4\alpha\lambda(kv - \mu) + 4\mu kv + 4\mu\lambda \\ \iff \mu^2 + \lambda^2 + k^2v^2 + 2\lambda(kv - \mu) - 2\mu kv &\geq 4\alpha\lambda(kv - \mu) \end{aligned} \quad (1)$$

(i) Si $kv - \mu > 0$, tenemos que $4\alpha\lambda(kv - \mu) \leq 4\lambda(kv - \mu) \quad \forall \alpha \in [0, 1]$

$$\begin{aligned} \mu^2 + \lambda^2 + k^2v^2 + 2\lambda(kv - \mu) - 2\mu kv &\geq 4\lambda(kv - \mu) \\ \iff \mu^2 + \lambda^2 + k^2v^2 - 2\lambda(kv - \mu) - 2\mu kv &= \left((kv - \mu)^2 - \lambda \right)^2 \geq 0 \\ \implies \mu^2 + \lambda^2 + k^2v^2 + 2\lambda(kv - \mu) - 2\mu kv &\geq 4\lambda(kv - \mu) \geq 4\alpha\lambda(kv - \mu) \end{aligned}$$

(ii) si $kv - \mu \leq 0$, tenemos que $4\alpha\lambda(kv - \mu) \leq 0 \quad \forall \alpha \in [0, 1]$

$$\begin{aligned} \mu^2 + \lambda^2 + k^2v^2 + 2\lambda(kv - \mu) - 2\mu kv &= \left((kv - \mu)^2 + \lambda \right)^2 \geq 0 \\ \implies \mu^2 + \lambda^2 + k^2v^2 + 2\lambda(kv - \mu) - 2\mu kv &\geq 0 \geq 4\alpha\lambda(kv - \mu) \end{aligned}$$

De las partes (i) y (ii) concluimos que se cumple (1), y entonces $\gamma_1, \gamma_2 \in \mathbb{R} \quad \forall \alpha \in [0, 1]$. \square

Apéndice B. Escenarios del sistema conmutado

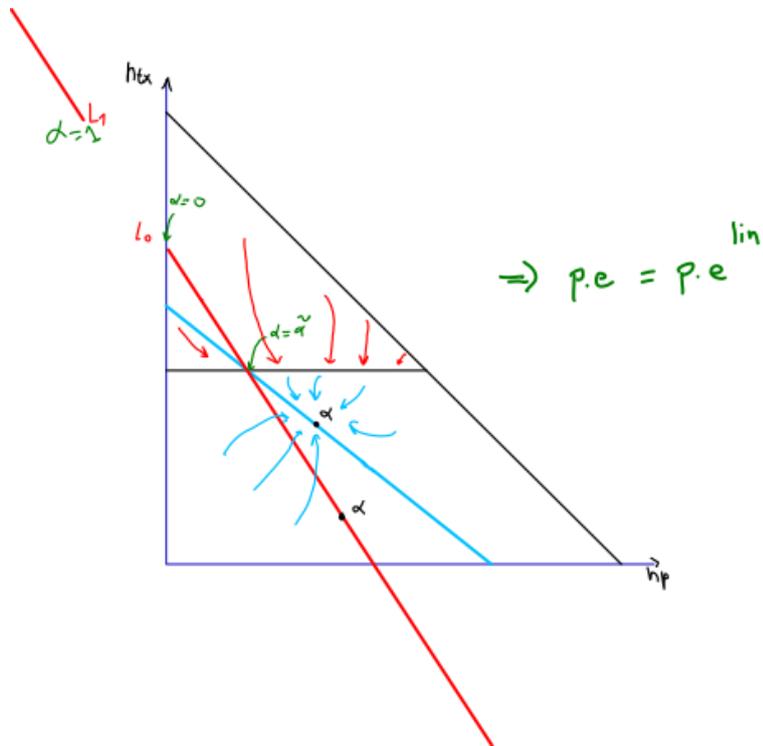


Figura B.2: Escenario 2 con $\lambda > k_1 v$, $\lambda > \frac{cv}{1-c/k}$, $\alpha > \tilde{\alpha}$ y $\alpha > \alpha_p$

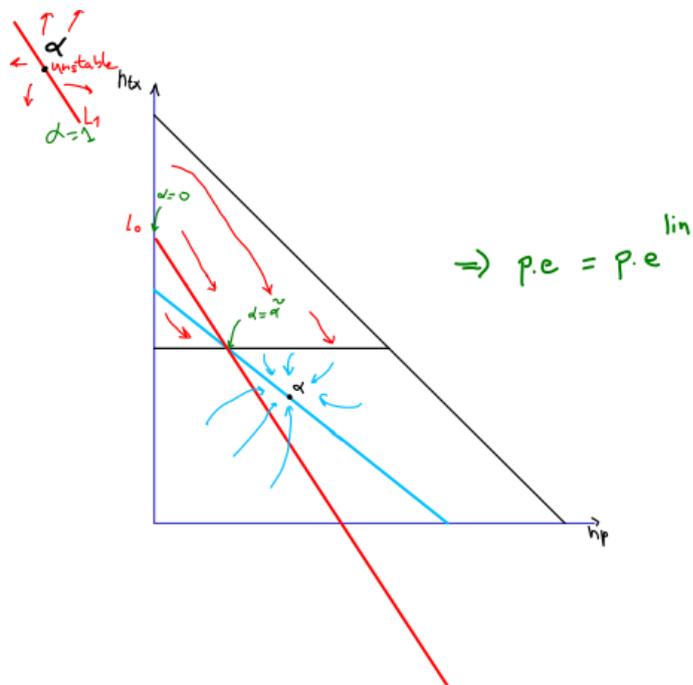


Figura B.3: Escenario 3 con $\lambda > k_1 v$, $\lambda > \frac{cv}{1-c/k}$, $\alpha > \tilde{\alpha}$ y $\alpha > \alpha_p$

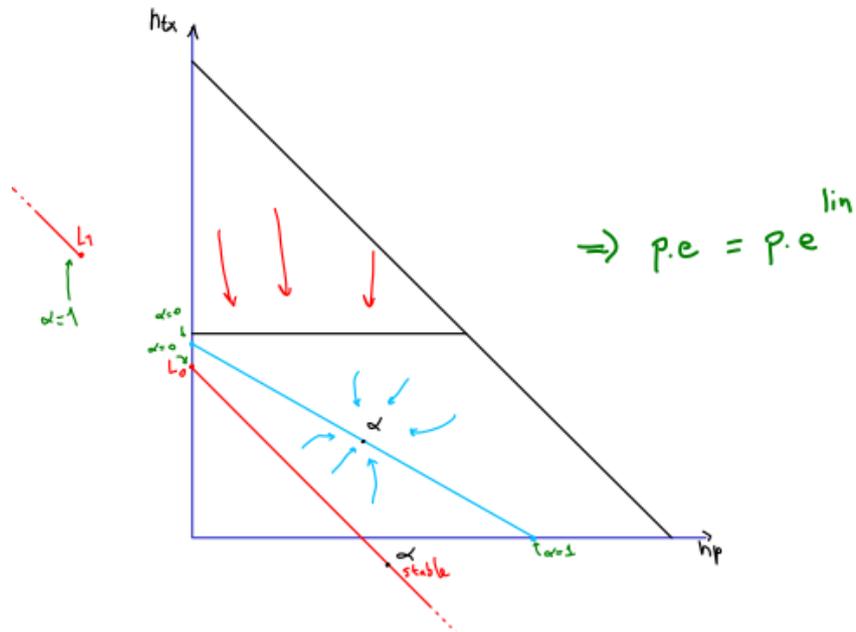


Figura B.4: Escenario 4 con $\lambda > k_1 v$, $\lambda < \frac{cv}{1-c/k}$ y $\alpha < \alpha_p$

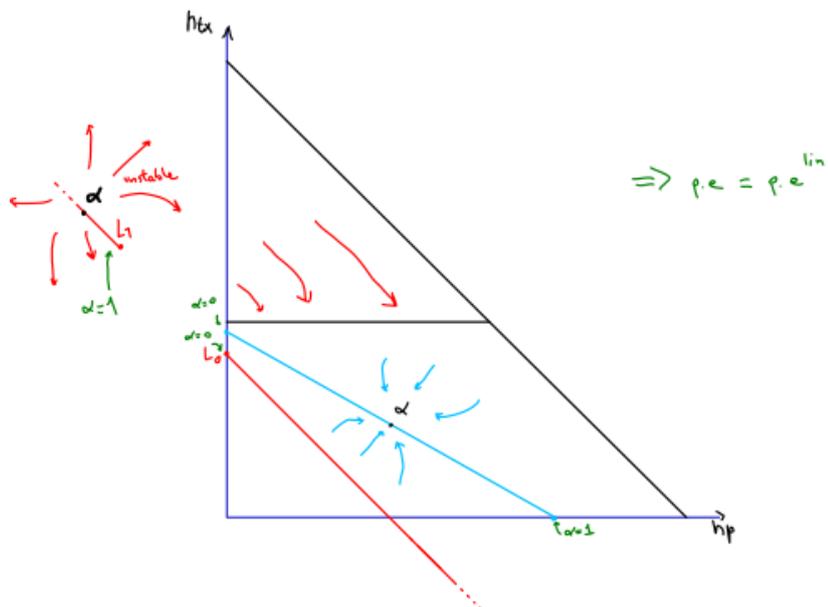


Figura B.5: Escenario 5 con $\lambda > k_1 v$, $\lambda < \frac{cv}{1-c/k}$ y $\alpha > \alpha_p$

Apéndice B. Escenarios del sistema conmutado

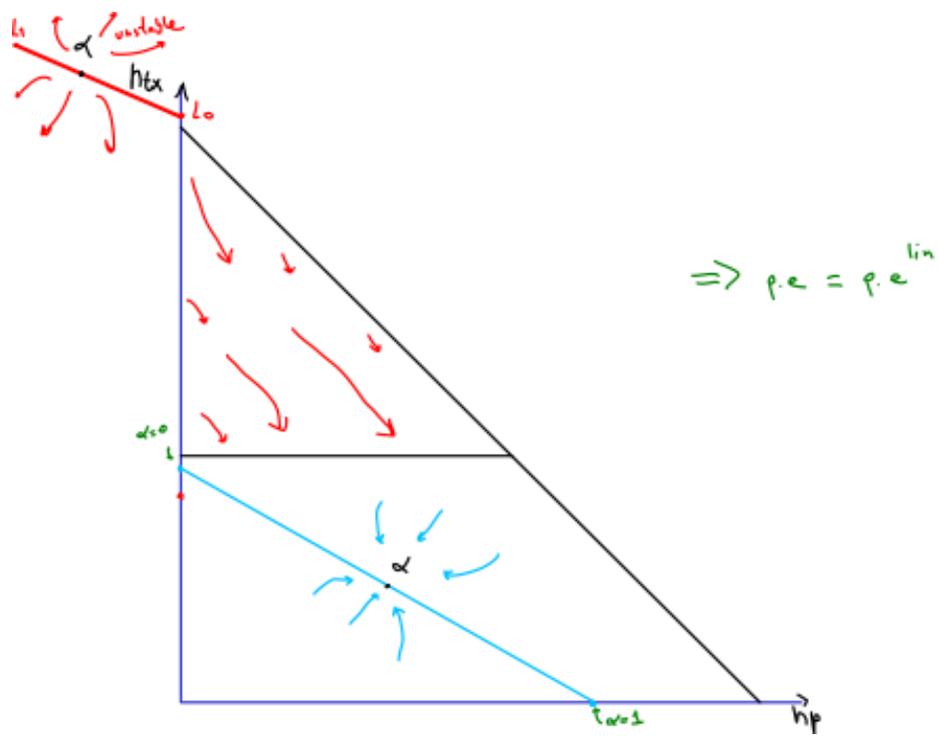


Figura B.6: Escenario 6 con $\lambda < k_1 v$

Apéndice C

Simulador del proceso estocástico

A los efectos de simular el proceso estocástico con la dinámica del proceso propuesto y verificar la aproximación de límite fluido, se implementó un simulador en Python.

Los modelos presentados en los capítulos 3 y 4 se simulan con una arquitectura que se resume a continuación.

En primer lugar se generan eventos de tareas y eventos de energía (en el modelo con energía), de forma que sean procesos Poisson con los parámetros correspondientes. Cada evento de tarea o de energía se asocia con cierto dispositivo IoT.

Un aspecto crucial al agregar un nuevo evento a la sucesión de eventos es ordenarlo temporalmente en la lista de eventos. Esto es clave ya que al procesar un evento, este puede desencadenar otro evento que debe ser atendido en el momento oportuno.

Los eventos de la lista de eventos son atendidos uno a la vez, y según el tipo de evento se procede a realizar la acción correspondiente. Esto va desde incrementar la energía de un nodo IoT, comenzar a procesar o transmitir una tarea o dar por perdida una tarea por estar el nodo ocupado o no disponer de energía.

Cada vez que se procesa un evento se actualizan las variables que registran las poblaciones de nodos de cada clase (procesando, transmitiendo y en espera con sus respectivas unidades de energía).

El código se encuentra disponible en el repositorio:

<https://gitlab.fing.edu.uy/gbelcredi/fog-iot>

Esta página ha sido intencionalmente dejada en blanco.

Referencias

- [1] E. Nygren, R. K. Sitaraman, and J. Sun, “The Akamai network: a platform for high-performance internet applications,” *ACM SIGOPS Operating Systems Review*, vol. 44, pp. 2–19, Aug. 2010.
- [2] CISCO, “Cisco visual networking index: Forecast and trends 2017–2022,” 2019. URL: <https://www.cisco.com/c/enius/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>.
- [3] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” Tech. Rep. NIST Special Publication (SP) 800-145, National Institute of Standards and Technology, Sept. 2011.
- [4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—a key technology towards 5g,” *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.
- [5] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, “A survey on computation offloading modeling for edge computing,” *Journal of Network and Computer Applications*, vol. 169, p. 102781, Nov. 2020.
- [6] T. G. Kurtz, “Limit theorems for sequences of jump Markov processes approximating ordinary differential processes,” *Journal of Applied Probability*, vol. 8, pp. 344–356, June 1971. Publisher: Cambridge University Press.
- [7] P. Robert, *Stochastic Networks and Queues*. Stochastic Modelling and Applied Probability, Berlin Heidelberg: Springer-Verlag, 2003.
- [8] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Feb. 2014.
- [9] S. M. Ross, *Stochastic Processes*. Wiley, 1996. Google-Books-ID: ImUPA-QAAMAAJ.
- [10] J. R. Norris, *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge: Cambridge University Press, 1997.
- [11] S. Ethier and T. Kurtz, *Markov Processes: Characterization and Convergence*. Wiley Series in Probability and Statistics, Wiley, 2009.

Referencias

- [12] C. Rattaro, L. Aspirot, E. Mordecki, and P. Belzarena, “Qos provision in a dynamic channel allocation based on admission control decisions,” *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 5, Feb. 2020.
- [13] O. Gil, *Curso introductorio a las ecuaciones diferenciales*. 1994.
- [14] G. Strang, *Differential equations and linear algebra*. 2014. OCLC: 887935097.
- [15] G. Salzer, “Poincare diagram: Classification of phase portraits,” 2020. <https://texample.net//tikz/examples/poincare/>.
- [16] L. Farina and S. Rinaldi, *Positive linear systems: theory and applications*. Pure and applied mathematics, New York: Wiley, 2000.
- [17] C. Meyer, S. f. Industrial, and A. Mathematics, *Matrix Analysis and Applied Linear Algebra*. EngineeringPro collection, Society for Industrial and Applied Mathematics, 2000.
- [18] S. Boyd, *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, 1st edition ed., Mar. 2004.
- [19] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, “Internet of things (iot) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5g-iot scenarios,” *IEEE Access*, vol. 8, pp. 23022–23040, 2020.
- [20] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh, “A Survey on the Computation Offloading Approaches in Mobile Edge/Cloud Computing Environment: A Stochastic-based Perspective,” *Journal of Grid Computing*, vol. 18, pp. 639–671, Dec. 2020.
- [21] M. Masdari and H. Khezri, “Efficient offloading schemes using Markovian models: a literature review,” *Computing*, vol. 102, pp. 1673–1716, July 2020.
- [22] B. Liu, Q. Zhu, W. Tan, and H. Zhu, “Congestion-Optimal WiFi Offloading with User Mobility Management in Smart Communications,” Aug. 2018. ISSN: 1530-8669 Pages: e9297536 Publisher: Hindawi Volume: 2018.
- [23] L. Chen, S. Zhou, and J. Xu, “Computation Peer Offloading for Energy-Constrained Mobile Edge Computing in Small-Cell Networks,” *IEEE/ACM Transactions on Networking*, vol. 26, pp. 1619–1632, Aug. 2018.
- [24] T. Q. Dinh, Q. D. La, T. Q. S. Quek, and H. Shin, “Learning for Computation Offloading in Mobile Edge Computing,” *IEEE Transactions on Communications*, vol. 66, pp. 6353–6367, Dec. 2018. Conference Name: IEEE Transactions on Communications.
- [25] C. Shannon, “Communication in the Presence of Noise,” *Proceedings of the IRE*, vol. 37, pp. 10–21, Jan. 1949. Conference Name: Proceedings of the IRE.

- [26] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [27] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice Hall, 1992. Google-Books-ID: FfpSAAAAMAAJ.
- [28] A. Tanenbaum, D. Wetherall, and A. Elizondo, *Redes de computadoras*. Pearson Educación, 2012.
- [29] M. S. Branicky, *Studies in hybrid systems : modeling, analysis, and control*. Thesis, Massachusetts Institute of Technology, 1995. Accepted: 2005-08-17T18:10:11Z.
- [30] M. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, pp. 475–482, April 1998.
- [31] G. G. Yin and C. Zhu, “Stability of Switching ODEs,” in *Hybrid Switching Diffusions: Properties and Applications* (G. G. Yin and C. Zhu, eds.), Stochastic Modelling and Applied Probability, pp. 217–250, New York, NY: Springer, 2010.
- [32] J. Oehlerking, *Decomposition of stability proofs for hybrid systems*. phd, Universität Oldenburg, 2011.
- [33] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

Esta página ha sido intencionalmente dejada en blanco.

Índice de tablas

| | |
|---|----|
| 3.1. Listado y definición de parámetros. | 25 |
| 3.2. Parámetros de la simulación 3.6.1 | 42 |
| 3.3. Valor medio del proceso estocástico vs punto de equilibrio del modelo fluido | 43 |
| 3.4. Parámetros de la simulación 3.6.2 | 43 |
| 3.5. Soluciones del problema de optimización | 43 |
| 3.6. Multiplicadores de Lagrange del problema de optimización | 46 |
| 3.7. Resultados de QoS | 46 |
| 4.1. Intensidades de nodos con carga baja | 52 |
| 4.2. Intensidades de nodos con carga intermedia ($j = 2, \dots, E - 1$) | 52 |
| 4.3. Intensidades de nodos completamente cargados | 53 |
| 4.4. Parámetros de la simulación | 56 |
| 4.5. Punto de equilibrio ODE vs Simulación Estocástica variando N | 58 |
| 4.6. Parámetros de simulaciones en escenarios de carga y descarga. | 62 |
| 4.7. Valores de equilibrio de variables agregadas del experimento de carga. | 67 |

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

| | |
|---|----|
| 1.1. Tráfico global de Internet, años 2017 a 2022 [2]. | 2 |
| 2.1. Cola M/M/1 como CTMC | 9 |
| 2.2. Aproximación fluida del Ejemplo 2.2.1 con parámetros $N = 500$, $\rho = 1.2$ y $\kappa = 1/7$ | 12 |
| 2.3. Diagrama de Poincaré. Adaptado de [15] | 14 |
| 2.4. Función convexa | 16 |
| 3.1. Arquitectura de offloading de tres capas (Local-Fog-Cloud). | 20 |
| 3.2. $R(\mathcal{L})$, throughput agregado en función de la cantidad de dispositivos transmitiendo | 22 |
| 3.3. Throughput vs Carga de distintos protocolos de acceso al medio [28] | 23 |
| 3.4. Escalado del throughput agregado de la red con N | 24 |
| 3.5. Regiones del sistema conmutado | 27 |
| 3.6. Una trayectoria del sistema conmutado queda definida por la se- cuencia de índices y las dinámicas de los sistemas. En este ejem- plo, la secuencia de conmutación es $[(1, t_0), (2, t_1), (1, t_2)]$. El siste- ma conmutado comienza con la dinámica del Sistema I y conmuta en $t = t_1$ al Sistema II. Luego en $t = t_2$ la trayectoria ingresa a la región inferior por lo que el sistema conmuta por última vez al Sistema I. Finalmente la trayectoria converge hacia un punto de equilibrio estable en la región inferior. | 28 |
| 3.7. Lugar geométrico del punto de equilibrio del Sistema I variando α . \mathbf{C}^{lin} se muestra en azul y corresponde a un segmento de recta incluido en el triángulo factible. | 30 |
| 3.8. \mathbf{C}^{sat} en dos escenarios distintos. El segmento de recta en rojo co- rresponde al lugar geométrico del punto de equilibrio con $\alpha \in [0, 1]$ cuando $k_1 v > \lambda$, y en verde el lugar geométrico cuando $k_1 v \leq \lambda$. . . | 32 |
| 3.9. Distintos escenarios posibles según los parámetros del sistema . . . | 35 |
| 3.10. En este escenario, ambos puntos de equilibrio son estables y se en- cuentran ubicados en la parte superior del triángulo. Todas las tra- yectorias convergen al punto de equilibrio del Sistema II | 36 |
| 3.11. En este escenario el punto de equilibrio del Sistema II es inestable y se ubica fuera del triángulo factible, todas las trayectorias convergen al punto de equilibrio del Sistema I | 37 |

Índice de figuras

| | |
|---|----|
| 3.12. Costo de offloading hacia el fog y el cloud. | 40 |
| 3.13. Evolución temporal de las densidades de población. Las trayectorias estocásticas de cada clase se compara con la correspondiente aproximación de límite fluido. Se muestran dos simulaciones con valores distintos de factor de offloading. | 44 |
| 3.14. Diagrama de fase con las trayectorias del ODE, la simulación estocástica y las restricciones de QoS. | 45 |
| 3.15. Evolución de la calidad de servicio en el Experimento I | 47 |
| 3.16. Tasa de pérdidas vs Tiempo de retardo. | 48 |
| 4.1. Evolución temporal de nodos en espera según nivel de carga. | 56 |
| 4.2. Evolución temporal de nodos transmitiendo según nivel de carga. | 57 |
| 4.3. Evolución temporal de nodos procesando según nivel de carga. | 57 |
| 4.4. Ubicación de los valores propios de F y A para el sistema especificado en Tabla 4.4. | 60 |
| 4.5. Ubicación de los valores propios de F y A con $E = 24$, y con el resto de los parámetros especificados en Tabla 4.4. | 60 |
| 4.6. Evolución temporal de nodos en espera sin energía, nodos procesando y transmitiendo. | 63 |
| 4.7. Evolución de las variables agregadas en escenario de descarga. | 64 |
| 4.8. SoC de red IoT en escenario de descarga. | 65 |
| 4.9. Porcentajes de pérdidas en escenario de descarga (a) totales (b) por energía (c) por ocupación. | 66 |
| 4.10. Densidad de nodos en espera según nivel de carga. | 67 |
| 4.11. Densidad de nodos transmitiendo según nivel de carga. | 67 |
| 4.12. Densidad de nodos procesando según nivel de carga. | 68 |
| 4.13. Evolución temporal de nodos en espera sin energía, nodos procesando y transmitiendo. | 68 |
| 4.14. Evolución de las variables agregadas en escenario de carga. | 69 |
| 4.15. SoC(t) | 70 |
| 4.16. Porcentaje de pérdidas en escenario de carga (a) totales (b) por ocupación (c) por energía. | 71 |
| B.1. Escenario 1 con $\lambda > k_1 v$, $\lambda > \frac{cv}{1-c/k}$ y $\alpha < \tilde{\alpha}$ | 79 |
| B.2. Escenario 2 con $\lambda > k_1 v$, $\lambda > \frac{cv}{1-c/k}$, $\alpha > \tilde{\alpha}$ y $\alpha > \alpha_p$ | 80 |
| B.3. Escenario 3 con $\lambda > k_1 v$, $\lambda > \frac{cv}{1-c/k}$, $\alpha > \tilde{\alpha}$ y $\alpha > \alpha_p$ | 80 |
| B.4. Escenario 4 con $\lambda > k_1 v$, $\lambda < \frac{cv}{1-c/k}$ y $\alpha < \alpha_p$ | 81 |
| B.5. Escenario 5 con $\lambda > k_1 v$, $\lambda < \frac{cv}{1-c/k}$ y $\alpha > \alpha_p$ | 81 |
| B.6. Escenario 6 con $\lambda < k_1 v$ | 82 |

Esta es la última página.
Compilado el lunes 18 julio, 2022.
<http://iie.fing.edu.uy/>