



Universidad del Bío-Bío  
Facultad de Ciencias Empresariales  
Magister en Ciencias de la Computación

# **Algoritmos de broadcasting para un LBS consciente de la privacidad de ubicación de sus usuarios**

Tesis para optar al grado de Magister en Ciencias de la Computación

Pablo Torres Osses  
paitorre@egresados.ubiobio.cl

**Profesor Co-Guía**  
Patricio Galdames  
pgaldames@ubiobio.cl

**Profesor Guía**  
Cristian Durán  
crduran@ubiobio.cl

4 de mayo de 2020

## Resumen

Los servicios basados en la ubicación (conocidos como LBS) son utilizados por usuarios de redes móviles inalámbricas y les dan a conocer, por ejemplo, detalles de los puntos de interés turísticos cercanos a la posición de un visitante, o bien para dar a conocer si en la vecindad de una persona existe algún hospital. El problema es que la ubicación se transforma en un pseudo-identificador cuando es relacionada con datos públicos ya que permite individualizar o crear un perfil del usuario. Hoy en día, la técnica más popular propuesta para proteger la privacidad de ubicación de una persona que accede a un LBS, consiste en que el usuario le envía una consulta con posición encubierta (CUE). Una CUE consiste de una consulta de rango tradicional pero asociada no solo a una sino que a  $k$  posiciones distintas ( $k$ -anonimato) y posibles de ser la verdadera posición del usuario (de hecho una de ellas es la real). El desafío por parte del LBS es cómo procesar eficientemente una cantidad masiva de consultas con posición encubierta sin que esto le signifique una mayor carga de trabajo, una mayor latencia y una mayor pérdida de la privacidad de ubicación de sus usuarios. Para enfrentar estos desafíos, nuestra idea es que el LBS procese en batch todas las consultas recibidas en un periodo de tiempo y ordene las respuestas apropiadamente para luego enviarlas en un broadcast público que sea escuchado por todos los usuarios, es decir, tanto por aquellos que solicitaron algún servicio como por aquellos que pretenden solicitarlo. De esta forma, el LBS proactivamente busca reducir la latencia, proteger la privacidad de ubicación de sus usuarios y reducir su carga de trabajo sirviendo a los usuarios antes que estos le envíen sus consultas. En este informe de tesis se proponen tres diferentes algoritmos de scheduling para ordenar los datos a transmitir en un broadcast: algoritmos de envergadura, algoritmo de popularidad, algoritmo de relevancia y otras tres variantes que incorporan la latencia. Se realizaron extensas simulaciones para evaluar la efectividad de estas técnicas en términos del rendimiento del LBS y la calidad de las respuestas transmitidas. Este trabajo también propone una métrica para evaluar el grado de pérdida de privacidad de ubicación cuando un usuario no encuentra respuesta en el broadcast y debe, por lo tanto, enviar su consulta al LBS.

# Índice general

<b>1. Introducción</b>	<b>3</b>
1.1. Hipótesis . . . . .	8
1.2. Objetivos de la investigación . . . . .	8
1.2.1. Objetivo general . . . . .	8
1.2.2. Objetivos específicos . . . . .	9
<b>2. Estado del arte</b>	<b>10</b>
2.1. Técnicas de protección de la privacidad de ubicación . . . . .	10
2.2. Construcción de un broadcast . . . . .	14
2.3. Contexto general de las técnicas de scheduling . . . . .	15
2.4. Técnicas de scheduling para consultas en demanda . . . . .	17
2.5. Técnicas de scheduling para consultas multi-ítem en demanda . . . . .	21
2.6. Propuestas de solución de las técnicas de scheduling . . . . .	23
2.7. Medición del rendimiento de las técnicas de scheduling . . . . .	27
<b>3. Alcances de la investigación</b>	<b>29</b>
3.1. Arquitectura del sistema . . . . .	29
3.1.1. Consultas con ubicación encubierta (CUEs) . . . . .	30
3.1.2. Ciclo de procesamiento y difusión de datos . . . . .	33
3.1.3. Broadcast . . . . .	34
3.1.4. Métrica de Privacidad de Ubicación . . . . .	35
<b>4. Estrategias de scheduling propuestas</b>	<b>37</b>
4.1. Antecedentes previos . . . . .	37
4.2. Algoritmo de scheduling basado en la envergadura . . . . .	40
4.3. Algoritmo basado en la popularidad . . . . .	45
4.4. Algoritmo basado en la relevancia . . . . .	51
4.5. Algoritmos de ajuste para las estrategias de scheduling . . . . .	58

<b>5. Evaluación del sistema</b>	<b>61</b>
5.1. Ambiente de pruebas . . . . .	63
5.2. Tiempo de respuesta . . . . .	65
5.2.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule . . . . .	66
5.3. Utilidad y efectividad . . . . .	76
5.3.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule . . . . .	77
5.4. Justicia . . . . .	83
5.4.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule . . . . .	83
5.5. Obsolescencia . . . . .	90
5.5.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule . . . . .	90
5.6. Privacidad . . . . .	94
5.6.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule . . . . .	94
5.7. Carga de trabajo . . . . .	97
5.7.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule . . . . .	97
<b>6. Conclusiones y trabajos futuros</b>	<b>100</b>
6.1. Recapitulación . . . . .	102
6.2. Trabajos futuros . . . . .	103
<b>Referencias</b>	<b>105</b>

# Capítulo 1

## Introducción

De acuerdo a William Stallings (Stallings, 2017), la privacidad conlleva proteger al usuario contra el descubrimiento y el uso indebido de su identidad y detalles de su persona. La especificación *common criteria* (Mellado, Fernández-Medina, y Piattini, 2007) incluye un conjunto de requerimientos funcionales para cumplir con la privacidad. Los criterios establecidos son el anonimato, seudónimo, desvinculación y la no observabilidad.

El anonimato y el seudónimo establecen garantías homólogas con la diferencia que el anonimato desprecia completamente el uso de una identidad. Por un lado, el anonimato implica que un usuario no da indicios de ella, y por otro, el seudónimo entrega un alias. En el caso de la desvinculación, ésta garantiza al usuario que pueda usar múltiples recursos o servicios sin que éstos puedan ser vinculados. Por último, la no observabilidad garantiza que un usuario pueda usar un recurso o servicio sin que otros, especialmente terceros, puedan observar el recurso o servicio que se está utilizando. Particularmente, la ubicación o paradero de las personas es un atributo que hoy en día puede ser vinculado con otros datos, con los cuales se puede concluir la identidad de quién está en un determinado lugar. Esto es posible dado que existe la tecnología y acceso a una gran cantidad de datos públicos que permiten correlacionar la identidad de las personas con su actuar (Sweeney, 2002).

Hay un sinnúmero de servicios para redes inalámbricas, entre ellos, los denominados servicios basados en la ubicación (LBS, según la nomenclatura anglosajona) que requieren de la posición exacta de un usuario para entregar una respuesta coherente. Un ejemplo de uso de estos servicios es la búsqueda del hospital más cercano para un usuario que ha sufrido un accidente o determinar el número de personas que ingresan o salen de un hospital. El primer ejemplo corresponde a un servicio que busca los  $k$  (en el ejemplo  $k = 1$ ) vecinos más cercanos y el segundo corresponde a una consulta de rango. Diversos trabajos (Galdames y Cai, 2012) señalan que una consulta de vecinos más cercanos puede transformarse en una

consulta de rango, por lo tanto, en este trabajo se asume que todas las consultas basadas en la ubicación corresponden a consultas de rango.

Todos los servicios basados en la ubicación suponen que el usuario está dispuesto a liberar su posición exacta, ya que presumen que el LBS es confiable y que éste dará un uso adecuado a la información obtenida. Sin embargo, en la práctica observamos que este supuesto de confianza no ha sido garantizado por las empresas que ofrecen estos servicios. En (Press, 2004) se ha conocido de diversos casos donde la ubicación de los usuarios no ha sido tan solo usada para identificarlos sino también para dañarlos. Por lo tanto, la posición de un usuario se convierte en un atributo que debe ser protegido, lo cual se denomina como *privacidad de ubicación*.

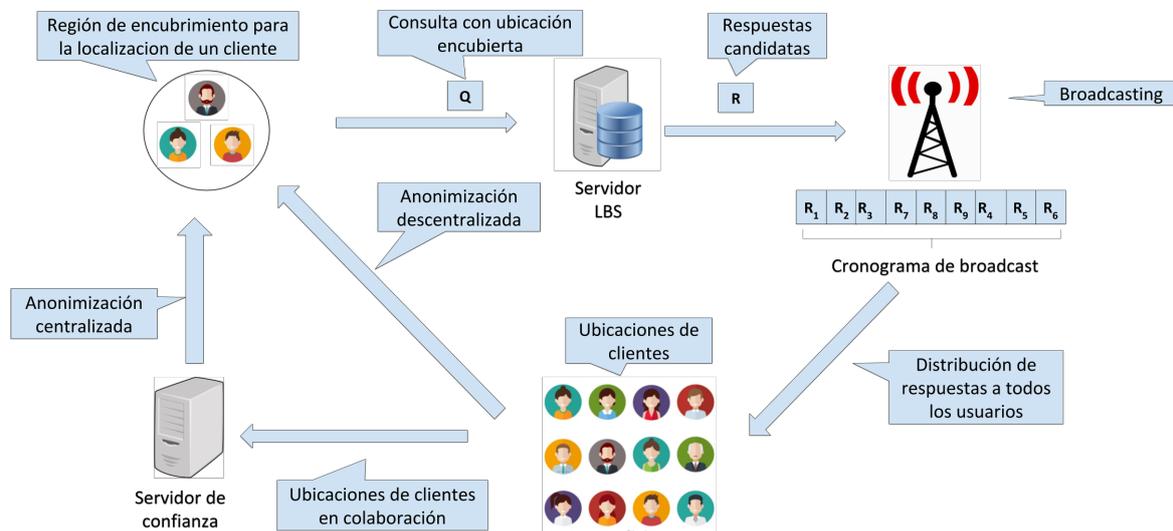
En la literatura se han propuesto diversas técnicas para garantizar la protección a la privacidad de ubicación tales como: 1) técnicas basadas en criptografía (Ghinita, Kalnis, y Skiadopoulos, 2007), 2) técnicas basadas en transformaciones espaciales (Indyk y Woodruff, 2006), 3) técnicas basadas en la perturbación de la ubicación (Hua, Tong, Xu, y Zhong, 2017; Andrés, Bordenabe, Chatzikokolakis, y Palamidessi, 2012) y 4) técnicas basadas en la reducción de la resolución de la ubicación (Gruteser y Grunwald, 2003; Chow, Mokbel, y Liu, 2006; Mokbel, Chow, y Aref, 2006; Chow, Mokbel, y He, 2008; T. Xu y Cai, 2009a; Alabdulatif, Kumarage, Khalil, y Yi, 2017). Estas últimas técnicas construyen una *región de encubrimiento* basada en el concepto de *k*-anonimato (Gruteser y Grunwald, 2003). Aquí se busca encontrar un área geográfica que encierre *k* posiciones distintas, una de las cuales es la real del usuario y las otras posiciones corresponden a otros posibles lugares en los que también se pueda encontrar el usuario.

El contexto tradicional de un LBS asume que existe un gran número de usuarios que, mediante dispositivos móviles inalámbricos, desean enviar hasta un servidor LBS confiable, ubicado en la Internet tradicional, un gran número de consultas de rango basadas en su localización exacta. Sin embargo, en este trabajo se asume que el servidor LBS no es confiable, en el sentido que una consulta de rango tradicional entrega información de una localización específica, la cual le permitirá al LBS relacionar varias consultas de rangos y así concluir con el comportamiento de un usuario específico. Es por ello que en este trabajo se asume que los usuarios ocultan sus posiciones exactas mediante regiones de encubrimiento. Se llamará a las consultas de rango que son contenidas por una región de encubrimiento como **consultas con ubicación encubierta (CUE)**. Esto se propone para diferenciar las CUEs de las consultas de rango tradicionales que se definen con respecto a una única posición, que es la ubicación real de su usuario/dueño. Luego que las CUEs son procesadas, se asume que el mismo servidor LBS se encarga de transmitir las respuestas en un broadcast a la red

inalámbrica donde están escuchando los usuarios del LBS. Para que el LBS y los usuarios operen como ha sido señalado se requieren resolver diversos desafíos:

1. **Desafíos en el procesamiento de CUEs en el servidor LBS:** Cuando un usuario oculta su ubicación exacta en una región de encubrimiento y envía una CUE al servidor LBS, éste deberá resolver la consulta considerando que cualquier posición de la región de encubrimiento es una ubicación válida en la que se podría encontrar el usuario. Por lo tanto, el procesamiento de las CUEs aumentará la carga de trabajo del servidor. Esto porque por cada CUE, el servidor deberá procesar  $k$  consultas de rango, de las cuales solo una de ellas es la real.
2. **Desafíos en la planificación de los datos a transmitir en un broadcast:** Ya que el servidor LBS debe procesar  $k$  consultas para resolver una CUE, éste debería intentar transmitir primero aquella respuesta que resulte ser la más probable de ser la correcta y así reducir el tiempo de respuesta a una CUE. Como esto resulta imposible (ya que las  $k$  consultas son igualmente probables de ser la real), el servidor LBS podría procesar por lotes un grupo de CUEs que estén a la espera de ser respondidas. La idea es apostar por aquellas respuestas que tengan mayor oportunidad de ser requeridas por alguien y, por lo tanto, darles mayor prioridad al momento de realizar el broadcast. Entonces, el desafío resulta en determinar una adecuada métrica que determine tal oportunidad.
3. **Desafíos por el lado del usuario al escuchar un broadcast:** Primero, dado que el servidor LBS transmitirá una gran cantidad de respuestas al usuario, éste deberá escuchar y filtrar los elementos de datos que le son útiles para responder a su consulta. Este proceso puede requerir del consumo excesivo de la limitada capacidad de energía del dispositivo móvil de cada usuario. Segundo, el usuario deseará limitar el envío de CUEs al LBS, dado que cada transmisión inalámbrica puede poner en riesgo su privacidad de ubicación.
4. **Desafíos por el lado de la privacidad de ubicación:** Si bien el usuario protege su ubicación encubriéndola (con otras  $k - 1$  posiciones), sin duda la mejor opción es que el servidor pueda predecir las consultas del usuario antes que éstas sean recibidas por el LBS. Lograr una predicción acertada permitirá al LBS obtener varios beneficios importantes. Primero, mejora la privacidad del usuario ya que evita que él libere su consulta al LBS y segundo, el servidor reduce su carga de trabajo ya que debe procesar menos consultas.

En este trabajo se propone enfrentar estos desafíos estableciendo una solución que opere sobre la arquitectura que se muestra a continuación en la **Figura 1.1**



**Figura 1.1:** Arquitectura del sistema de broadcasting

Esta arquitectura acepta dos maneras de proporcionar una CUE al LBS. La primera manera permite que múltiples usuarios envíen sus posiciones exactas y sus consultas de rango a un *servidor de anonimato*, también conocido como servidor de confianza, el cual construye una región de encubrimiento ( $k - 1$  posiciones altamente probables de ser la posición real del usuario) para cada CUE con el fin de proteger la privacidad de la ubicación de su dueño. La otra forma permite que los mismos usuarios puedan rechazar el uso del servidor de anonimato, y que estos deban de colaborar entre si para construir regiones de encubrimiento. En este trabajo se asume que ambas formas de construir una CUE son igualmente válidas y que los usuarios son libres de escoger una o ambas.

Cuando el servidor LBS recibe un grupo de CUEs, éste las procesa en batch y ordena sus respuestas para luego difundirlas a través de un broadcast que es escuchado por todos los usuarios de la red inalámbrica. Para procesar las CUEs existen un sin número de trabajos realizados (Papadias, Zhang, Mamoulis, y Tao, 2003) y no es tema de estas tesis abordar su procesamiento eficiente, por lo que cualquier técnica seleccionada pudiera ser igualmente válida. Lo que si es importante en esta tesis es planificar el orden de transmisión de las respuestas, el cual se determina mediante la ejecución de diversos algoritmos de scheduling o planificación propuestos en este trabajo. Estos algoritmos buscan encontrar un ordenamiento que reduzca principalmente la carga de trabajo del servidor y limite los tiempos de respuesta.

En esta misma línea, la solución considera los siguientes aspectos principales:

1. Ya que el servidor LBS procesa periódicamente por lotes un grupo de CUEs en espera de ser resueltas, este servidor podría tomar como ventaja que consultas geográficamente cercanas tenderán a presentar respuestas muy similares, por lo tanto ésta similitud pudiera ser considerada para determinar si una respuesta tiene mayor probabilidad de ser realmente requerida por alguien y así reducir el tiempo de respuesta al planificar su broadcast (Galdames y Cai, 2012).
2. Los elementos de datos son transmitidos en un broadcast público a todos los usuarios del sistema, ya sea que hayan o no enviado consultas al servidor LBS. Esto se decide con el objetivo de que algunos usuarios puedan evitar el envío de una CUE si los datos que son difundidos en un broadcast responden a sus consultas de rango. Esto último obliga a los usuarios a participar activamente en el procesamiento de sus consultas.
3. Para evaluar el rendimiento del sistema se utilizan métricas tradicionales de evaluación como el tiempo de respuesta, la carga de trabajo y la pérdida de privacidad de ubicación. Para medir esto último, se propone una métrica para determinar esta pérdida al someter su consulta al LBS.

Sin perjuicio de lo señalado con anterioridad también se decide incluir otros aspectos secundarios a investigar en esta tesis:

1. Deseamos investigar qué tan efectivo puede resultar realizar una planificación de broadcast de datos al medir otras métricas de rendimiento (además de la carga de trabajo, privacidad y el tiempo de respuesta) tales como la injusticia, obsolescencia, efectividad y la utilidad.
2. La posibilidad de reutilizar una misma planificación de transmisión de datos en varios broadcast, para así reducir aún más la carga de trabajo del LBS. Consideramos que este reuso es posible ya que es probable que los usuarios se mantengan durante un lapso de tiempo en una misma zona geográfica y, por lo tanto, sus siguientes consultas serán relativas al mismo lugar geográfico en el que se encuentran. En este sentido si el LBS no recibe consultas, éste podría concluir que los datos transmitidos en los broadcast son relevantes para los usuarios y su repetición es conveniente. En el caso contrario, si el LBS comienza a recibir un gran número de consultas, es porque el broadcast no responde a lo esperado por los usuarios y, por lo tanto, se requiere una nueva planificación.

Considerando todos los desafíos y características del escenario de aplicación, en este trabajo de tesis se proponen **tres diferentes algoritmos de planificación o scheduling denominados como: envergadura, popularidad y relevancia**. Se estudian y comparan su efectividad en balancear las diversas métricas mencionadas para un grupo de usuarios móviles.

## 1.1. Hipótesis

Existe un algoritmo de scheduling que ordena las respuestas a consultas bajo diversos criterios que permitan balancear las métricas de carga de trabajo, tiempo de respuesta, injusticia, obsolescencia, la utilidad de las respuestas y la pérdida de privacidad de ubicación.

## 1.2. Objetivos de la investigación

En esta sección se presenta los objetivos generales y específicos de la investigación.

### 1.2.1. Objetivo general

El objetivo general de esta investigación es diseñar e implementar diversas técnicas de scheduling para determinar el orden de transmisión de respuestas a un grupo de CUEs en un broadcast. Estas técnicas de scheduling buscan equilibrar diversas métricas de rendimiento tales como:

1. *Carga de trabajo del servidor LBS*. El número de consultas procesadas por el servidor LBS por unidad de tiempo.
2. *La obsolescencia de las respuestas del broadcast*. Debido al movimiento de los usuarios, la respuesta a la consulta original ya no sea válida para la nueva posición del usuario.
3. *La latencia que experimenta cada usuario mientras espera por los datos que responden su consulta*. La latencia se refiere al tiempo de respuesta promedio que experimenta cada usuario.
4. *La injusticia en los tiempos de respuesta obtenidos cuando un grupo de usuarios espera por sus respuestas en el broadcast*. Que tan desigual es el tiempo de respuesta de un usuario al compararlo con el obtenido por otros usuarios.

5. *La pérdida de la privacidad de ubicación de los usuarios.* Cuándo el usuario debe consultar al LBS o el broadcast para resolver una consulta.

### **1.2.2. Objetivos específicos**

Los objetivos específicos de la investigación son:

1. Realizar el estudio bibliográfico sobre técnicas de scheduling empleadas para difundir datos en redes inalámbricas, que genere un balance adecuado entre privacidad, injusticia, carga de trabajo, entre otras métricas.
2. Desarrollar tres técnicas de scheduling (técnica de envergadura, popularidad y relevancia) para la construcción un broadcast que difunda elementos de datos para los usuarios de un servidor LBS.
3. Proponer la definición de una métrica de privacidad de ubicación para un LBS que realiza broadcasting de datos.
4. Desarrollar experimentos sobre las técnicas de scheduling para demostrar el cumplimiento del objetivo general de la tesis.

# Capítulo 2

## Estado del arte

En este capítulo se presentan diversos trabajos relacionados con algoritmos de scheduling. Se inicia el estudio presentando las principales técnicas para proteger la privacidad de la ubicación. Luego se realiza un estudio sobre las técnicas para procesar consultas que tienen la ubicación encubierta. Se continúa con las diversas técnicas para construir un broadcasting. También se hace una especificación de cada una de las técnicas de scheduling, las cuales se catalogan en técnicas para consultas en demanda y consultas multi-item. Finalmente, se analizan las diferentes técnicas mostradas y cómo se mide el rendimiento de las mismas con la finalidad de tener una base para analizar las propuestas de solución presentadas en este trabajo.

### 2.1. Técnicas de protección de la privacidad de ubicación

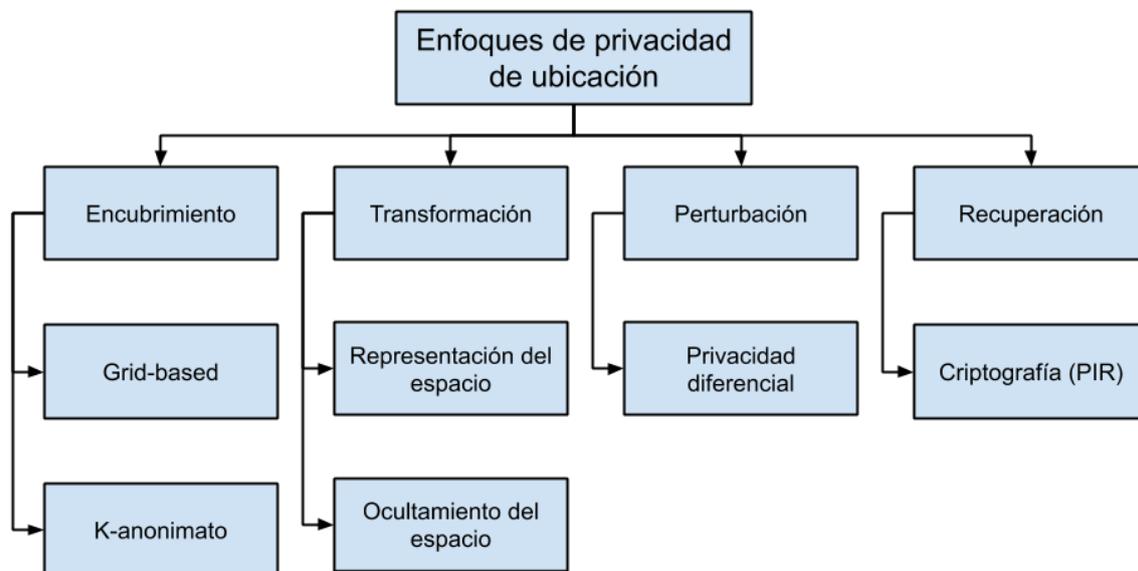
En la literatura (Krumm, 2009; Riboni, Pareschi, y Bettini, 2009; Toch, Wang, y Cranor, 2012) se puede observar que los servidores LBS son un grupo particular de los servicios conscientes del contexto (CAS). Los CAS son sistemas de solicitudes de información que conocen datos como la localización, trayectorias, edad, género, entre otros. En otros términos, todo tipo de atributos que se utilicen para entregar información personalizada a los usuarios. Así se puede justificar el hecho de compartir información sensible a un servidor LBS, dado que existen usuarios que buscan información personalizada según su localización. En (Krumm, 2009; Riboni y cols., 2009; Toch y cols., 2012) se hace especial énfasis en las preocupaciones de privacidad en los diferentes contextos en que se usa un LBS, por ejemplo, un atacante obtiene acceso a los datos de ubicación y los utiliza para descubrir, mediante diversas técnicas, el paradero de un sujeto y todo tipo de información relacionada a la misma. Krumm (Krumm, 2009), en su trabajo hace especial énfasis en el estudio de los peligros de privacidad que nacen

a partir de la ubicación publicada por los usuarios a un LBS. Los peligros más preocupantes son los ataques de inferencia y combinación. Estos ataques consisten en identificar a una persona mediante el rastro que deja de sus ubicaciones en combinación con datos extraídos de fuentes públicas. Riboni (Riboni y cols., 2009) en su estudio de la literatura hace énfasis sobre el problema de la privacidad relacionado con las técnicas de asociación sensible entre la identidad del usuario y su información considerada sensible. Riboni dice que el control de acceso en los sistemas conscientes del contexto tiene como prioridad restringir la entidad que realiza solicitudes en Internet, es decir, utilizar servidores de anonimato. Touch (Toch y cols., 2012) explica que el comportamiento del sistema consciente del contexto vulnera la privacidad de los usuarios. Por ejemplo, Facebook comparte la información de su red social para que empresas como eBay, Travelocity y The New York Time puedan personalizar la información que es transmitida a los usuarios. Para describir la evolución que han tenido los diversos inconveniente de privacidad, se puede observar que entre el año 2003 y 2009 en la mayor parte de los trabajos se explica que el método para poder identificar a los usuarios era a través de la asociación entre la identidad y la información sensible, la cual, no era masiva en aquel periodo, sin embargo, desde el año 2012 la información sensible comienza a ser extraída desde las redes sociales y diversas fuentes de información pública. Se entiende por información sensible a toda aquella que entrega la posibilidad de individualizar a una persona.

Barkhuus y Dey (Barkhuus y Dey, 2003) realizaron un estudio sobre las preocupaciones de privacidad en dos tipos de LBS: 1) servicios basados en el seguimiento de la ubicación y 2) servicios basados en el reconocimiento de la ubicación exacta. Para ser precisos, allí se define la privacidad de ubicación de la siguiente forma: la ubicación de una persona es un atributo específico de su identidad, como el nombre y el número de seguro social (para nosotros el rut), donde la principal diferencia entre la ubicación y la mayoría de los atributos es que la ubicación cambia continuamente y es relevante cuando es usada en el contexto de dispositivos móviles. De este modo, es posible distinguir el comportamiento según los movimientos espaciales.

Para resguardar o anonimizar la privacidad de la ubicación de los usuarios, en la literatura se proponen diversos enfoques. Existen técnicas que utilizan regiones de encubrimiento para hacer anónima la ubicación del usuario, las cuales requieren especificar un  $k$  – anonimato, donde  $K$  corresponde al número de usuarios dentro de una región de encubrimiento (Chow y cols., 2006; Mokbel y cols., 2006; Chow y cols., 2008; T. Xu y Cai, 2009a; Barkhuus y Dey, 2003; Ma, Zhou, y Yang, 2015; Kini y Kulkarni, 2017; J. Wang y cols., 2017; L. Liu, 2007; Pan, Xu, y Meng, 2011; T. Xu y Cai, 2009b; Gupta, Muralidharan, Gogri, y Damodhiran, 2010;

Chow, Mokbel, y Aref, 2009; Bamba, Liu, Pesti, y Wang, 2008; Li y Zhu, 2012; Lee, Oh, Kim, y Chang, 2012; Alizadeh-Shabdiz, Jones, Morgan, y Shean, 2007; Kalnis, Ghinita, Mouratidis, y Papadias, 2007). Las técnicas de k-anonimato presentan diversas desventajas cuando existe una gran cantidad de información pública. Es por eso que se utilizan técnicas que mejoran el k-anonimato, como por ejemplo, las técnicas de l-diversidad (Xue, Kalnis, y Pung, 2009) y m-invarianza (Dewri, Ray, Ray, y Whitley, 2010a), donde las técnicas de l-diversidad mejora el k-anonimato cuando se requiere información individual y grupal de usuarios, mientras que la técnica de m-invarianza se enfoca en mejorar el k-anonimato cuando el usuario requiere información especializada; observar Encubrimiento en la **Figura 2.1**. Um y otros autores (Um, Kim, Lee, Jang, y Chang, 2012; Jang, Jang, y Chang, 2012) crean las regiones de encubrimiento usando diagramas de Voronoi para encontrar los k-vecinos más cercanos. También, Ngo y otros (Ngo y Dang, 2013) propusieron cambiar el procesamiento de regiones de encubrimiento con forma de rectángulo por polígonos y círculos. Por otro lado, hay técnicas que utilizan la indistinguibilidad geoespacial con el objetivo de proteger la posición exacta de un usuario a través de la incorporación de ruido (noisy)(Andrés y cols., 2012; Gruteser y Grunwald, 2003); observar Perturbación en la **Figura 2.1**. Por el lado de la criptografía, algunos autores que realizan sus investigaciones basados en sistemas que recuperan respuestas cifrada desde un servidor LBS, las cuales satisfacen una consulta espacial cifrada (Ghinita y cols., 2007; Ghinita, Kalnis, Khoshgozaran, Shahabi, y Tan, 2008; Magkos, 2012; Khoshgozaran, Shirani-Mehr, y Shahabi, 2013; Paulet, Kaosar, Yi, y Bertino, 2013); observar Recuperación en la **Figura 2.1**. Finalmente, están las técnicas de transformación espacial, las cuales buscan interpretar la ubicación de modo que no puedan ser distinguidas o interpretadas(Indyk y Woodruff, 2006); observar Perturbación en la **Figura 2.1**.



**Figura 2.1:** Taxonomía de 4 tipos de enfoques para anonimizar la ubicación: a) Técnicas de encubrimiento, b) técnicas de transformación, c) técnicas de perturbación y d) técnicas recuperación criptográfica.

En este trabajo de tesis se asume que la técnica del *k-anonimato* es la empleada para proteger la privacidad de ubicación cuando un usuario necesita acceder a un servidor LBS. Esto porque es una de las soluciones más citadas en la literatura actual dada su versatilidad de ser aplicada a diversos escenarios prácticos, ya sea en aquellos en los cuales los usuarios son móviles y/o estáticos o bien cuando los puntos de interés son móviles (preguntas sobre otros usuarios) y/o estáticos (preguntas sobre hoteles, hospitales) también. Si bien la criptografía ha desarrollado técnicas promisorias para proteger la privacidad de ubicación, aún no es escalable computacionalmente al ser empleada en escenarios móviles. En el caso de las técnicas basadas en la privacidad diferencial, por un lado esta solución entrega una respuesta alterada por un ruido (ruido de Laplace) que busca proteger la privacidad, la cual podría ser interpretada de manera similar al resultado que logra el *k-anonimato*, por otro lado no es aún una técnica tan madura como el *k-anonimato* y es actualmente tema de investigación en como esta se aplica en distintos escenarios. Si bien este trabajo de tesis no persigue el desarrollo de una técnica para construir consultas encubiertas, ésta requiere ser conocida de antemano para evaluar la efectividad de los algoritmos de planificación propuestos en esta tesis. Como trabajo futuro, se podría realizar un estudio comparativo del *k-anonimato* y privacidad diferencial para el contexto de aplicación de este trabajo de tesis.

## 2.2. Construcción de un broadcast

En este trabajo difundimos las respuestas como Liu (F. Liu, Hamza-Lup, y Hua, 2010) a través de un canal de broadcast que ayuda a proteger la privacidad de ubicación. Sin embargo, aprovechamos la redundancia espacial de las consultas para construir un broadcast que tiene un grado de utilidad a través del tiempo. Según Galdames (Galdames y Cai, 2012) es posible balancear cuatro métricas de rendimiento de un broadcast:

1. Tiempo de respuesta del usuario.
2. Utilidad y efectividad del broadcast.
3. Justicia del broadcast.
4. Carga de trabajo del servidor.

El tiempo de respuesta es el transcurso de tiempo entre el momento en que se crea una consulta hasta que se responde de forma efectiva. La utilidad y efectividad del broadcast sirve para medir que tan productivo el trabajo que realiza el servidor para satisfacer a los usuarios. La justicia del broadcast es la capacidad que éste tiene para entregar respuestas equánimes en función al stretch y el jitter de un broadcast. El stretch ecuanimizará un broadcast cuando los usuarios reciban sus respuestas según el número que requieren, mientras que el jitter lo ecuanimizará cuando las respuestas sean recibidas en lapsos similares; en los próximos capítulos se detallan mas estos dos conceptos. Finalmente, la carga de trabajo es el esfuerzo que hace el servidor para responder una determinada cantidad de consultas tan eficientemente como le sea posible.

El problema con estas métricas es que se evaluaron con distintos broadcast, ya que los algoritmos de scheduling, hasta el año 2012 eran volátiles, es decir, para cada difusión de respuestas se crea/construye/planifica un nuevo broadcast, por lo tanto, se asume que las consultas son distintas a las anteriores y las respuestas de los anteriores broadcast son desechadas. Para solucionar aquello, en este trabajo buscamos que el broadcast sea modificado o reconstruido en vez de crear uno nuevo. Dado lo anterior, se observa que además de aprovechar la redundancia espacial, también se puede aprovechar la redundancia temporal. Para ser específicos, en este trabajo se construyen los algoritmos de scheduling basados en la redundancia espacial y temporal, y se evalúan con las 4 métricas expuestas anteriormente, además de incorporar dos nuevas métricas: ***la obsolescencia del broadcast*** y ***la privacidad de la ubicación***.

Lu (Lu, Wu, Li, y Pan, 2016) construye un broadcast con respuestas que maximizan la justicia de la difusión emitida a los usuarios y maximiza el número de consultas respondidas. Sin embargo, Zaixin asume que toda la información es relevante al momento de ser difundida, a diferencia de nuestro trabajo, en donde se debe asumir que hay respuestas que son irrelevantes por el hecho de usar regiones de encubrimiento. La utilidad absoluta de las respuestas, en el trabajo de Zaixin, se debe a que se procesan consultas comunes (no espaciales) que no son conscientes del contexto (por ejemplo, ubicación) y de la privacidad de la ubicación.

### 2.3. Contexto general de las técnicas de scheduling

Recientemente existe un creciente interés en el desarrollo de aplicaciones móviles en tiempo real y de la mano crece el número de usuarios móviles. Las aplicaciones y los usuarios requieren que la información fluya en lapsos cortos de tiempo. Es por ello que el broadcasting se hace necesario. Actualmente hay tres formas básicas de realizar data broadcast:

1. Transmisión basada en push (Acharya, Franklin, y Zdonik, 1995).
2. Transmisión basada en pull (Acharya y Muthukrishnan, 1998).
3. Transmisión híbrida (Guo, Das, y Pinotti, 2001).

La difusión basada en push, también llamada difusión estática o periódica, se caracteriza por:

1. El servidor difunde datos periódicamente, es decir, el broadcast está predefinido para difundir un determinado tipo de información.
2. El usuario no solicita datos.
3. La probabilidad de acceso a los datos son calculados a partir de distintas técnicas que se basan en el estudio de patrones de acceso.
4. Existe una alta cantidad de datos que ayuda a predecir los datos a difundir.
5. Los datos difundidos se repiten en forma periódica para los usuarios de broadcast. Esto se debe a que las técnicas basadas en push asumen que el patrón de acceso calculado, es constante a través del tiempo.

Por otra parte, la difusión basada en pull, también llamada difusión dinámica o en demanda, tiene cuatro características:

1. El servidor difunde datos dinámicamente, es decir, el broadcast se va modificando a medida que cambian las consultas.
2. El usuario solicita datos.
3. Los datos a difundir son ordenados con distintas técnicas que se basan en las solicitudes pendientes realizadas por los usuarios.
4. No existe información que ayude predecir los datos a difundir.
5. Los datos difundidos no se repiten en los usuarios de broadcast, ésto se debe a que se asume que las solicitudes de los usuarios varían a través del tiempo, por lo tanto, una vez que se transmite el broadcast se debe calcular uno nuevo para las nuevas solicitudes pendientes.

Finalmente, la difusión híbrida (Guo y cols., 2001) realiza una combinación tanto de la técnicas push como pull. Esta combinación se caracteriza por:

1. El servidor difunde datos tanto periódicamente como dinámicamente.
2. Se analiza el patrón de acceso y las solicitudes de los usuarios.
3. Las solicitudes son divididas en dos grupos:
  - a) Populares.
  - b) Menos populares.
4. Se calcula un punto de corte entre las consultas populares y menos populares.
5. Los datos difundidos a través del broadcast de transmisiones, divididos en populares y menos populares, son asignados de dos formas:
  - a) Los datos populares son transmitidos de forma periódica (basado en push).
  - b) Los datos menos populares son transmitidos de forma dinámica (basados en pull).

En este trabajo es de interés estudiar las técnicas de planificación tales como las basadas en pull y las técnicas híbridas. Ésto se debe a que la naturaleza del problema se sitúa sobre una serie de factores que deben ser considerados:

1. Los usuarios realizan solicitudes de elementos de datos.
2. Los usuarios esperan que los elementos de datos solicitados sean transmitidos prontamente.
3. El broadcast se renueva completa o parcialmente a medida que llegan nuevas consultas.

Así, se puede establecer que el enfoque basado en push, el cual asume que el patrón de acceso es constante en el tiempo y además repite los elementos datos dentro del broadcast, no representa la naturaleza del problema estudiado en este trabajo.

A modo de conclusión, en este trabajo se utilizan técnicas enfocadas en la planificación pull. De modo que, los usuarios envían sus consultas a un servidor para obtener respuestas.

## 2.4. Técnicas de scheduling para consultas en demanda

**Técnica de planificación LTSF (longest total stretch first):** Acharya y otros investigadores (Acharya y Muthukrishnan, 1998) propusieron un modelo que se separa de las anteriores investigación de solicitudes en demanda. Esto se debe a que consideran que los usuarios hacen solicitudes de tamaños heterogéneos y los usuarios acceden a la información de forma dinámica. Por otro lado, para evaluar una consulta en vez de medir su tiempo de respuesta se mide el rendimiento con una métrica llamada stretch; la cual se define como la razón entre el tiempo de respuesta y el tiempo de servicio. La heurística de planificación plantea un algoritmo llamado primero el stretch total más largo (LTSF) para optimizar el stretch y lograr un equilibrio entre el caso más desfavorable y el caso promedio. Una implementación de LTSF no es práctica para un sistema grande, ya que en cada broadcasting, el servidor tiene que recalcular el stretch para cada elemento de datos que son solicitados, y por lo tanto, el algoritmo de schedule se convierte en un cuello de botella debido a la alta tasa de consulta requeridas.

**Técnica de planificación RxW (Request number x Wait time):** Askoy y Franklin (Askoy y Franklin, 1999) proponen un modelo que considera la planificación en un ambiente escalable y en demanda. Esta vez las solicitudes son consideradas con un tamaño homogéneo, sin embargo, se mantiene el concepto de que los usuarios acceden a la información de forma dinámica. Se propone una métrica llamada RxW donde R corresponde al número de solicitudes que son respondidas y W al tiempo de la solicitud que más tiempo lleva pendiente. Aquí se proponen tres heurísticas, la primera busca de forma exhaustiva el broadcast con mayor RxW, la segunda incorpora un recorte al espacio de búsqueda y la tercera que compensa la escalabilidad en favor del tiempo de respuesta en el caso peor y promedio.

**Técnicas de planificación EDF ( Earliest Deadline First) y LLF(Least Laxity First) :** En (Kalyanasundaram y Velauthapillai, 2003) se investiga sobre los algoritmos de scheduling en tiempo real. Esto significa que hay tres puntos a considerar: Primero, que el algoritmo tenga una complejidad temporal baja. Segundo, el algoritmo de planificación debe responder la mayor cantidad de solicitudes durante el periodo de transmisión de un broadcast. Y tercero, los elementos de datos solicitados tienen tamaños heterogéneos. Las métricas de evaluación que utiliza Kalyanasundaram son: 1) deadline y 2) laxity. Ambas métricas están relacionadas. Por un lado el deadline permite medir el plazo máximo en que los usuarios necesitan los elementos de datos de la solicitud; no obstante, los elementos de datos solicitados por los usuarios son distintos (en tamaño), por lo tanto, la métrica laxity permite adaptar el deadline según las características de los elementos de datos de una solicitud. En otras palabras, si un usuario solicita un elemento de datos que es muy grande, él debería estar dispuesto a que el sistema relaje (laxity) la prioridad con respecto al plazo límite estipulado, es decir, una solicitud de un elemento de dato grande con un plazo límite pequeño tiene mayor laxitud. Las métricas de evaluación propuesto son dos: 1) Bcast y 2) Bcast2; el primer algoritmos se basa en un deadline suave y el último en un deadline restrictivo.

**Técnica de planificación ACR (Aggregated Critical Requests) :** En (Fang, Vrbsky, Dang, y Ni, 2004) se presenta el concepto de tiempo restringido. Este tiempo restringido es distinto a los dos tipos de estrategias basadas en deadline, las cuales son: 1) soft deadline y 2) hard deadline. En un entorno de transmisión basado en tiempo restringido, el objetivo es determinar qué tan bien los algoritmos de scheduling garantizan que el servidor no pierda el deadline de una solicitud, en vez de minimizar el tiempo de espera de la solicitud. La gran diferencia que separa a la estrategia de tiempo restringido con otras, como las basadas en deadline, es que la primera actúa eficientemente en el contexto de solicitudes hechas por dispositivos móviles en tiempo real. El modelo del algoritmo de planificación tiene como entrada solicitudes en tiempo real, las cuales son realizadas sobre un solo elemento de dato y todos los elementos de datos tienen un tamaño homogéneo. Bajo el contexto del tiempo real el algoritmo de planificación debe ser dinámico, por lo tanto se asume que el servidor mantiene un registro de : 1) número de solicitudes por elemento de dato, 2) tiempo de espera de las solicitudes, 3) deadline de las solicitudes. Por otro lado, el algoritmo de planificación, denominado ACR, tiene como principal objetivo minimizar el número de plazos obsoletos del elemento de dato cuando este es transmitido en un determinado intervalo de tiempo.

**Técnica de planificación SIN- $\alpha$  (Slack time Inverse Number of pending requests) :** En (J. Xu, Tang, y Lee, 2006) se estudia el tiempo restringido de solicitudes en tiempo real y en demanda. Todo dentro del contexto de servicios basados en la ubicación de usuarios

móviles. Jianliang hace un especial énfasis en que las solicitudes deben tener un deadline para rendir eficientemente en tiempo real y en demanda. El modelo de transmisión utilizado en [Jianliang, 2006] tiene las siguientes características: Primero, son difundidos los elementos de datos de solicitudes que están pendiente. Segundo, se considera el deadline asociado con cada una de las solicitudes. Tercero, una solicitud nueva puede ser unida con una que está pendiente solo si el mismo elemento de dato es solicitado. Cuarto, los elementos de datos tiene un tamaño fijo.

**Técnica de planificación HUD (Highest Utility First)** : En (Dewri, Ray, Ray, y Whitley, 2008) se caracteriza el broadcasting por estar presente en entornos ubicuos. El cual abarca la comunicación peer-to-peer y usuario-servidor. El hecho es que el uso dinámico y escalable de los datos que requieren estos dos escenarios es la razón por la que múltiples solicitudes para el mismo elemento de dato, deberían ser atendidas por una sola transmisión. Por otro lado, los elementos de datos a difundir están ligados con una determinada calidad del servicio (QoS). En donde se busca cumplir con dos objetivos: 1) maximizar las solicitudes atendidas en una transmisión de datos y 2) minimizar el tiempo de espera de un elemento de datos requerido por una solicitud. Para esto, Dewri establece que el deadline y la prioridad están relacionadas con la utilidad de los elementos de datos a través del tiempo. Esto quiere decir que, dependiendo del plazo y la importancia de una solicitud, es posible calcular la utilidad de un elemento de dato a través del tiempo de forma global, es decir, dentro del conjunto de elementos de datos solicitados. Esto revela dos hechos importantes que se deben priorizar en la transmisión: 1) solicitud con el primer plazo a caducar primero (EDF, earliest deadline first) y 2) solicitud con la mayor utilidad primero (HUD, highest utility First).

**Técnicas de planificación MPCF (Maximum Paid Cost First) y MVGF (Maximum value Gained First)** : En (Lei, Vrbsky, y Xiao, 2010) se refieren al broadcasting como un sistema de difusión de elementos de solicitudes que satisface múltiples consultas. Por otro lado, se refiere al scheduler como un algoritmo que construye un broadcast con elementos de solicitudes para ser difundido a través de un broadcast. Tanto el broadcasting como el scheduling tienen como objetivo mejorar la escalabilidad, el tiempo de respuesta y la tasa de obsolescencia de solicitudes. Lei propone un modelo de sistema en donde asumen que: 1) las solicitudes son respondidas por elementos de datos homogéneos y 2) el usuarios de broadcast construido es de tipo mixto, esto quiere decir que se minimiza el número de solicitudes obsoletas para consultas en tiempo real y en tiempo no-real. Dado lo anterior, las métricas de evaluación señaladas son dos: 1) tiempo de respuesta y 2) deadline, en donde el objetivo es minimizar el tiempo de respuesta y minimizar el número de solicitudes obsoletas. Finalmente, Lei y otros autores propone dos estrategias de ordenamiento del broadcast: 1) costo máximo pagado

primero (MPCF) y 2) valor máximo ganado primero (MVGf).

**Técnica de planificación MDMP (Minimum Deadline Meet Probability) y MBSP (Maximum Bounded Slack Probability)** : En (Dewri, Ray, Ray, y Whitley, 2010b) se analiza el problema de la transmisión de datos en tiempo real y descentralizados, es decir, datos que se ubican en distintos servidores. Se proponen dos heurísticas: el límite mínimo de tiempo límite (MDMP) y la máxima probabilidad de holgura limitada (MBSP). MDMP puede ser considerada una versión estocástica de EDF, donde el elemento de dato con mayor prioridad es aquel que tiene la menor posibilidad de que caduque su tiempo límite. MBSP es una extensión de MDMP, donde se asume un que las consultas tienen un tiempo límite mínimo, de este modo, los elementos de datos con mayor prioridad son aquellos que tienen mayor probabilidad de ser respondidas antes del tiempo límite.

**Técnica de planificación LxRxW (Lost requests x Request number x Wait time)** : En (Hu, Fan, Luo, Peng, y Du, 2015) se analiza el problema de transmisión de elementos de datos en tiempo real y en demanda. Por otra parte, se analiza el problema de índices aéreos, de modo de reducir el consumo de energía de los dispositivos móviles que utilizan un broadcast. El algoritmo de planificación propuesto, llamado LxRxW, tiene como objetivo mejorar el tiempo de acceso promedio y la tasa de abandono de solicitudes (Request drop ratio). Es indicado como una mejora de SIN- $\alpha$ , ya que considera los puntos débiles. Por ejemplo, se asume que no todas las solicitudes son urgentes. La estrategia para planificar con LxRxW es encontrar el broadcast con el mínimo valor. Hu y otros autores señalan que para mejorar la tasa de abandono de solicitudes, se debe considerar tanto el tiempo de espera como el deadline de las solicitudes. Por otro lado, se propone un método de ajuste dinámico para la longitud de los índices aéreos (DAIL).

**Técnicas de planificación ES-LxRxW (Equals Split) y US-LxRxW (Unequals Split)** : En (Hu, Xia, Du, y Wu, 2015) se analiza el problema de transmisión de elementos de datos en tiempo real, en demanda y urgentes, además, se consideran elementos de datos de diferentes tamaños. Se dice que las estrategias de planificación como SIN- $\alpha$  y LxRxW cuentan con un funcionamiento superior a otros algoritmos que tienen los mismos objetivos. Sin embargo, existen deficiencias cuando son utilizados con consultas de tamaños diferentes. Dado lo anterior, se propone un pre procesamiento de las consultas, de modo que, las respuestas son divididas en sub respuestas de tamaños uniforme. La técnica de planificación base es LxRxW, la cual calcula la urgencia de una consulta. Se proponen dos tipos de división, ES (equals split) y US (unequals split). Donde ES tiene un criterio de división que permite ordenar las respuestas con un tamaño uniforme, por otro lado, US considera el ancho de banda en el que se transmite el broadcast con respuestas de tamaño diferente.

## 2.5. Técnicas de scheduling para consultas multi-ítem en demanda

**Técnica de planificación DTIU (Dynamic Temperature Inverse Urgency)** : En (Chen, Lee, y Liu, 2010) consideran que algoritmos de planificación tales como SIN- $\alpha$ , RxW y LWF, los cuales son usados para la transmisión en tiempo real, no son suficientes para la planificación de solicitudes de múltiples elementos. El principal problema observado es el de la incompletitud de solicitudes. Por ejemplo, para un usuario que solicita los datos R : A, B, C, D la solicitud es completa cuando se transmiten todos los elementos. Sin embargo, dado que los algoritmos SIN- $\alpha$ , RxW y LFW discriminan los elementos de datos por popularidad y urgencia, la mayor parte de la transmisión no completa los elementos de datos solicitados, ya que hay elementos de datos menos populares dentro de una misma solicitud que quedan obsoletos por no cumplir el plazo establecido. Para solucionar este problema, es propuesto el algoritmo de planificación DTIU (Temperatura Dinámica Urgencia Inversa). Por lo tanto, este algoritmo de planificación considera dos variables para el diseño de una heurística: 1) Popularidad (temperatura) de las solicitudes y 2) Urgencia de las solicitudes. El modelo del algoritmo de planificación tiene como entrada solicitudes dinámicas en tiempo real, de las cuales se conocen: 1) el momento de llegada y 2) el plazo establecido de obsolescencia. Entonces, a partir de una serie de cálculos, el algoritmo de planificación entrega la prioridad de la solicitud. Finalmente, el algoritmo de planificación DTIU tiene como principal objetivo reducir la tasa de obsolescencia de los elementos de datos menos populares para completar las solicitudes en tiempo real.

**Técnica de planificación  $BAT^d$  (NP-hard case of Bounded Access Time problem) y  $BAT^{d2}$  (Well-solvable NP-hard case of Bounded Access Time problem)** : En (J.-Y. Wang, 2012) proponen evaluar el peor tiempo de acceso de los datos difundidos a través de un canal inalámbrico. Wang asume que los usuarios solicitan varios elementos de datos a la vez, los cuales son de tamaño homogéneo. El principal objetivo de Wang es minimizar el peor tiempo de acceso promedio. Con este enfoque se puede garantizar que los elementos de datos menos populares son difundidos con la misma justicia con que se transmiten los elementos más populares. Por lo tanto, existe un equilibrio entre la temporalidad de la transmisión y la justicia de la difusión de los elementos de datos. Wang presenta cinco algoritmos de planificación basados en un conjunto de datos para el problema de acceso limitado (BAT). El problema BAT se resume en encontrar un conjunto de respuestas, tan grande como sea posible, de modo que su peor tiempo de acceso sea el mínimo posible. También, el problema BAT tiene dos variaciones las cuales son : 1)  $BAT^d$  y 2)  $BAT^{d2}$ . Estas variaciones profundizan

la resolución del problema NP-hard del tiempo de acceso limitado. Para el problema  $BAT^d$  y  $BAT^{d2}$  se proponen las siguientes heurísticas: 1) Las solicitudes con elementos de datos similares deberían asignarse a un mismo conjunto solicitudes, 2) Las consultas grandes deben asignarse aparte aunque tengan datos similares y 3) Las solicitudes pequeñas que no comparten elementos son asignadas en un conjunto de forma descendente.

**Técnica de planificación PVC (Profit Versus Cost)** : En (Lv, Lee, Li, y Chen, 2012) se trata el problema de solicitudes en demanda que deben ser respondidas en tiempo real. El modelo propuesto por Lv y otros autores asume que: primero, el tamaño de los elementos de datos es homogéneo. Segundo, las solicitudes arriban al servidor en demanda. Tercero, los usuarios pueden solicitar múltiples elementos de datos de una vez. Cuarto, tanto las solicitudes respondidas como las que tiene su plazo caducado son eliminadas de la cola de espera del servidor. Por último, el usuarios de broadcast es servido a través de un único canal. Sin perder la generalidad, las métricas de evaluación son tres: 1) beneficio (profit), 2) costo de oportunidad (opportunity cost) y 3) tiempo de holgura/relajo. Por un lado, el beneficio es calculado para una solicitud y significa que mientras más solicitudes requieren un elemento de dato y esas solicitudes requieren menos elementos, este último se hace más beneficioso para el broadcast. Por otro lado, el costo de oportunidad mide la utilidad de un elemento de dato al ser difundido en un determinado momento del broadcast. Entonces, dado lo anterior el algoritmo de scheduling propuesto es llamado PVC. En donde, el principal objetivo es maximizar el beneficio y la utilidad/costo de un elemento de dato de las solicitudes.

**Técnicas de planificación MTRS (Maximum Throughput Request Selection) y MLRO (Minimum Latency Request Ordering)** : En (Lu y cols., 2016) estudian el caso de la transmisión de múltiples elementos de datos para responder una consulta. El objetivo de esta investigación es minimizar el tiempo promedio de acceso del broadcast. Por otro lado, las respuestas deben ser respondidas en demanda y en tiempo real. Se proponen dos algoritmos de scheduling. Primero está MTRS que apunta a buscar el conjunto de respuestas, a consultas pendientes, que responde tantas consultas con la menor cantidad de respuestas o elementos de datos, para ello se utiliza una heurística. Finalmente está MRLO que busca el orden de respuestas (o elementos de datos) que minimiza la latencia de tiempo de acceso. En conclusión, Lu establece reglas para trabajar en base al orden en que se difunden las respuestas, demostrando que, a partir de distintas métricas, se puede encontrar distintos broadcast de broadcast.

## 2.6. Propuestas de solución de las técnicas de scheduling

Existen diversas técnicas de planificación para broadcasting de respuestas en demanda para bases de datos ordinarias. El enfoque de las técnicas varía según el contexto o fenómeno que se busca evaluar. El contexto de una técnica se define como la circunstancias en que opera un algoritmo de scheduling. Los contextos más comunes son:

1. Cantidad de elementos de datos que solicita una consulta:
  - a) Único elemento de dato.
  - b) Múltiples elementos de datos.
2. Tamaño de los elementos de datos.
3. Tiempo de respuesta de los elementos de datos.

A pesar de que en la literatura se asume que las técnicas para múltiples elementos de datos son mal atendidas por las técnicas de un único elemento de dato (Chen y cols., 2010), es posible que al cambiar el contexto en que se aplican las técnicas, sea posible planificar consulta espaciales anónimas. Las técnicas de scheduling abordarán diferentes temáticas, sin embargo, entre ellas existen objetivos comunes, tales como:

1. Objetivo 1: Optimizar el stretch.
2. Objetivo 2: Optimizar la carga de trabajo
3. Objetivo 3: Optimizar la tasa de abandono u obsolescencia.
4. Objetivo 4: Optimizar la utilidad.

El objetivo 1 está enfocado en que el servidor equilibre el tiempo de respuesta según el tamaño de las consultas. Por ejemplo, una consulta A solicita un elemento de dato I1 de tamaño 10 y una consulta B solicita un elemento de dato I2 de tamaño 20. Suponiendo que existen otras consultas que solicitan elementos de datos con tamaños entre 10 y 20. El objetivo 1 propone que, dependiendo del tiempo de llegada y el tamaño de la consulta, el elemento de dato tiene una determinada prioridad. Siguiendo con el ejemplo, se considera justo que el elemento de dato sea posicionado de modo que todas las consultas tengan un tiempo de espera tan igual como sea posible.

El objetivo 2 está enfocado en que el servidor responda a mayor cantidad de consultas en el menor tiempo posible. Asumiendo que todas las consultas son respondidas en un

broadcast, siempre existe una consulta que se responde finalizando la difusión. Por ejemplo, para un determinado conjunto de consultas es posible determinar que existe un determinado broadcast que en períodos cortos de tiempo responde la mayor cantidad de consulta posibles.

El objetivo 3 está enfocado en que el servidor responda la mayor cantidad de consultas antes de que queden obsoletas. Se asume que las consultas establecen un plazo límite de tiempo para ser respondidas. Por ejemplo, para un determinado grupo de consultas, es posible responder efectivamente en un tiempo corto una amplia cantidad de consultas antes de que las respuestas queden obsoletas.

Finalmente el objetivo 4 está enfocado en que el servidor responda las consultas con mayor utilidad. Se asume que el ambiente de solicitudes es altamente escalable. Por ejemplo, para un determinado grupo de consultas, se responde en el menor tiempo posible las más populares o más solicitadas.

Como se puede observar, los objetivos planteados anteriormente son equivalentes a los objetivos de esta investigación. Por lo tanto, las técnicas de scheduling correspondientes serán transformadas para resolver el problema de la planificación de CUEs.

Las técnicas de scheduling gozan de características propias del contexto en el que se basan. Por ejemplo técnicas para consultas de múltiples elementos de datos que son solicitados en tiempo real (Lv y cols., 2012). Sin embargo, las características resuelven sólo una parte del problema de la planificación de respuestas.

La técnica de planificación LTSP (Acharya y Muthukrishnan, 1998) demuestra que es posible planificar un broadcast considerando el tamaño de las consultas. Se asume que el tiempo de acceso es prioritario cuando las consultas son de diferentes tamaños. Sin embargo, la técnica de planificación LTSP no considera la planificación en tiempo real, ya que el proceso de ordenamiento es costoso.

La técnica de planificación RxW (Askoy y Franklin, 1999) demuestra que es posible mantener un equilibrio de tiempo de espera entre las consulta populares y las menos populares. Si bien ésta técnica considera que todas las consultas tienen el mismo tamaño, a diferencia de la técnica de planificación LTSP, es una técnica basada en la decisión para atender las consultas en un tiempo óptimo. Sin embargo, la técnica de planificación RxW no considera la tasa de abandono de las consultas y el cálculo del óptimo broadcast sigue siendo costoso.

Las técnicas de planificación EDF y LLF (Kalyanasundaram y Velauthapillai, 2003) demuestran que es posible establecer tiempos límites para las consulta; estos tiempo límites son permisivos, en el sentido que si los plazos de tiempo son sobrepasados, las consultas deben ser sancionadas pero no eliminadas de un broadcast. A pesar de todo lo anterior, se asume que todas las consultas tienen la misma importancia.

La técnica de planificación ACR (Fang y cols., 2004) demuestra que es posible minimizar la tasa de abandono y optimizar el costo computacional para calcular o construir un broadcast. En esta técnica se asume que las consultas que no están dentro del plazo no son puestas en el broadcast, de modo que se recibirá otra consulta en su reemplazo. Ésta técnica de planificación no considera el número de solicitud actual y el tiempo de espera de los datos.

La técnica de planificación SIN- $\alpha$  (J. Xu y cols., 2006) demuestra que el plazo límite de tiempo en que se puede responder una consulta no implica que un elemento de dato no sea difundido. Por lo tanto, a diferencia de la técnica de planificación ACR la prioridad se calcula para los elementos de datos como el tiempo de holgura inverso al número de consultas pendientes. Asumiendo que el elemento de dato es enviado mientras el plazo límite de tiempo, de todas las consultas que lo soliciten, aun no se cumple.

La técnica de planificación HUD (Dewri y cols., 2008) demuestra que es posible calcular la prioridad en base a la utilidad a través del tiempo para los elementos de datos. La utilidad se considera una función que se calcula como la razón entre un valor de prioridad y el tiempo de respuesta. De modo que, mientras más grande es el tiempo de respuesta, la prioridad disminuye. Por otro lado, la técnica de planificación reorganiza el broadcast cuando llegan nuevas consultas.

Las técnicas de planificación MDMP y MBSP (Dewri y cols., 2010b) demuestran la posibilidad de procesar un broadcast de forma descentralizada. El almacenamiento descentralizado de datos introduce un problema novedoso para los algoritmos de scheduling, específicamente si los elementos de datos deben recuperarse de un servidor de datos antes de la transmisión a gran escala. Por lo general, el algoritmo de scheduling tiene un conocimiento completo del tiempo requerido para difundir un elemento de dato y utiliza esta información para decidir un broadcast. Sin embargo, cuando los datos deben ser recuperados desde distintos servidores de datos, este conocimiento asume una forma estocástica.

La técnica de planificación LxRxW (Hu, Fan, y cols., 2015) demostró que la mayoría de los algoritmos de planificación tuvieron en cuenta el plazo límite de solicitud y se concentraron en cómo reducir la tasa de abandono de las solicitudes. Pero muy pocos algoritmos podrían optimizar el tiempo de espera del elemento de datos. Sin embargo, la técnica de planificación LxRxW toma en cuenta el número de solicitudes perdidas durante el siguiente periodo de difusión de respuestas, el número de solicitudes y el tiempo de espera.

Las técnicas de planificación ES-LxRxW (Hu, Xia, y cols., 2015) proponen un tamaño estándar de división equitativa en diferentes distribuciones para el plazo límite de tiempo y diferentes tamaños de elemento de datos, mientras que US-LxRxW propone un tamaño estándar de división para la distribución de diferentes tamaños de los elementos de da-

tos y diferentes algoritmos de scheduling. Se demostró que ambos pueden adaptarse a la planificación en tiempo real muy bien.

En las técnicas de planificación mostradas anteriormente, es posible encontrar consulta en tiempo real y tiempo no real. Estableciendo una línea divisoria entre las técnicas de planificación que construyen una broadcast una vez finalizada una difusión y las técnicas que modifican las prioridades en tiempo real.

Las técnicas de planificación MPFC y MVGF (Lei y cols., 2010) demuestran que es posible planificar en base al tiempo de respuesta y el número de vencimiento del plazo límite. Por otro lado, se demuestra que las técnicas mejoran el rendimiento del broadcast a pesar de la intensidad que cambia la carga de trabajo.

Por otro lado, las anteriores técnicas de planificación se basan en la idea de que las consultas solicitan un elemento de dato a la vez. Sin embargo, existen técnica de planificación que responden consulta de con múltiples elementos de datos y a la vez deben transmitir todos los elementos de datos en un broadcast.

La Técnicas de planificación DTIU (Chen y cols., 2010) demuestran que es posible planificar consultas de múltiples elementos con plazos límites de tiempo. Por otro lado, se incorpora el concepto de urgencia, el cual indica que una consulta con menos elementos de datos y con mayor tiempo de espera, tiene mayor prioridad. Finalmente, permite entregar todos los elementos de datos en un solo broadcast.

Las Técnicas de planificación  $BAT^d$  y  $BAT^{d2}$  (J.-Y. Wang, 2012) demuestra que es posible planificar las características del conjunto de consultas para optimizar el peor tiempo de acceso. Se asume que al crear conjuntos con un número fijo de consultas, al determinar el grupo con un largo máximo (respecto a la cantidad de elementos de datos solicitados), los usuarios que observan la difusión, nunca tendrán que esperar un mayor tiempo al establecido por ese largo máximo.

La Técnica de planificación PVC (Lv y cols., 2012) demuestra que el aporte de los elementos se debe calcular en base al grupo de elementos que solicita una consulta. Específicamente, es más rentable transmitir un elemento de datos que puede satisfacer más solicitudes o acercar las solicitudes a la finalización. Para cuantificar esta idea, proponen un nuevo concepto llamado profit, para caracterizar la contribución potencial de un elemento pendiente al rendimiento del sistema.

Las Técnicas de planificación MTRS y MLRO (Lu y cols., 2016) demuestran que es posible incorporar planificar consulta multi-elementos en base a criterios generales. Específicamente, un broadcast construido en base a la carga de trabajo y el tiempo de acceso para consultas multi-elementos.

De todo lo anterior se debe considerar la diferenciación que hay entre las métricas de scheduling para ordenar las respuestas en un broadcast y las métricas para evaluar el broadcast de respuestas. Esto permitirá que en los próximos capítulos no exista ambigüedad cuando se evalúen estos dos aspectos diferentes.

Acharya y otros autores (Acharya y Muthukrishnan, 1998) proponen la métrica llamada stretch. La cuál es la razón entre el tiempo de respuesta por el tiempo de servicio. Esta métrica permite evaluar los elementos de datos según su tamaño. En el capítulo 4 se explica el uso de la métrica.

Al igual que Liu y otros autores (F. Liu y cols., 2010), en esta investigación, las consulta espaciales encubiertas son divididas en consultas espaciales más específicas, sin embargo, las métricas de heurísticas no son utilizadas. Si bien en esta investigación todos los elementos de datos tienen el mismo tamaño, se incorpora el stretch para evaluar el tamaño de las CUEs según la cantidad de elementos de datos solicitados.

Al igual que Fang (Fang y cols., 2004), Dewri (Dewri y cols., 2008), Wang (J.-Y. Wang, 2012) y Jingsong (Lv y cols., 2012), en este trabajo se evalúan las métricas de heurísticas de la popularidad y la relevancia desde el punto de vista de la cantidad de solicitudes realizadas sobre un elemento de dato. La popularidad y relevancia se consideran dos métricas de heurísticas distintas. En el capítulo 4 se explica las diferencias entre la popularidad y la relevancia.

A modo de conclusión, se debe considerar que trabajos como (Galdames y Cai, 2012; Lu y cols., 2016; F. Liu y cols., 2010) son la base para establecer nuestra investigación enfocada en las técnicas de scheduling. Sin embargo, las técnicas de scheduling presentadas en este trabajo tienen el desafío de responder consultas que contiene elementos de datos irrelevantes.

## 2.7. Medición del rendimiento de las técnicas de scheduling

Según nuestro conocimiento, las mediciones del rendimiento pueden ser analizadas tanto desde el punto de vista del usuario como del servidor.

En (Acharya y Muthukrishnan, 1998) se mide el rendimiento de average response time (ART) y stretch. Analizar ART significa medir el tiempo promedio desde el inicio del broadcast hasta el momento que se responde una consulta. Por otro lado, analizar stretch significa medir la razón entre el response time y service time. El service time es el tiempo para responder una consulta como si fuera la única en ser respondida. Es posible observar que el rendimiento en ART no es el mas eficiente ya que la técnica se basa en optimizar el stretch.

En (Askoy y Franklin, 1999) se mide el rendimiento de average wait time (AWT). Analizar

AWT significa medir el tiempo promedio que un usuario espera en una cola de servicio antes de satisfacer su consulta.

En (Dewri y cols., 2008) se mide el rendimiento de la técnica de planificación HUD. Específicamente, es medido percentage utility (PU). Analizar PU significa medir el porcentaje de utilidad de los elementos de dato. La utilidad de los elementos de datos se calcula en función de la popularidad.

En (Chen y cols., 2010; Lv y cols., 2012; Hu, Fan, y cols., 2015; Hu, Xia, y cols., 2015; J. Xu y cols., 2006; Fang y cols., 2004; Lei y cols., 2010) se mide el rendimiento del deadline miss ratio (DMR). Analizar DMR significa medir la tasa de obsolescencia de las solicitudes en función de un deadline (tiempo límite) propuesto por el usuario o calculado por el servidor. El servidor puede estimar el deadline en base a datos histórico de las solicitudes.

En (Chen y cols., 2010) se mide el rendimiento de scheduling efficiency ratio (SER). Analizar SER significa medir el porcentaje de efectividad que tiene un broadcast para responder consultas de múltiples elementos. Esta medición es realizada para comprobar que las técnicas tradicionales no responden de forma óptima las consultas con deadline. Por otra parte, se mide el rendimiento de percentage of broadcast bandwidth used (PBBU). Específicamente se mide el porcentaje de ancho de banda de los elementos de datos con menos solicitudes o menos populares.

En (Lu y cols., 2016) se mide el rendimiento de access latency (AL). Analizar AL significa medir el tiempo transcurrido entre múltiples elementos de datos solicitados por una consulta. Esto se debe a que los elementos de datos responden a diferentes consultas en diferentes momentos.

En (J.-Y. Wang, 2012; Hu, Fan, y cols., 2015; Hu, Xia, y cols., 2015; Lei y cols., 2010) se mide el rendimiento de access time (AT). Analizar AT significa medir el tiempo transcurrido desde el momento en que se envía una consulta al servidor LBS hasta el momento en que se satisface por completo.

Dado lo anterior se debe considerar que en éste trabajo, para evaluar el rendimiento de las diferentes técnicas de schedule, se considera el tiempo de respuesta, utilidad y efectividad del broadcasting, justicia del broadcasting, obsolescencia de las respuestas, privacidad de los usuarios y carga de trabajo del servidor LBS.

# Capítulo 3

## Alcances de la investigación

En este capítulo se hace énfasis en los aspectos considerados de base para llevar a cabo esta investigación. Por ejemplo la arquitectura del sistema, las consultas con ubicación encubierta, los usuarios del servidor LBS y el broadcasting en el que se transmiten las respuestas. Además, presentaremos la métrica para medir la privacidad de ubicación.

### 3.1. Arquitectura del sistema

El escenario de aplicación emplea la arquitectura de la **Figura 1.1**, cuyas principales componentes se describen a continuación:

1. **Servidor LBS:** Este servidor divide el área de servicio en una grilla, en la que cada celda es de igual tamaño. Esta división del área es conocida por todos los usuarios del sistema de antemano, pero no así los puntos de interés situados en cada celda de la grilla. El LBS indexa las celdas con el fin de responder eficientemente qué puntos de interés son cubiertos por una consulta de rango.
2. **Usuarios móviles:** Son usuarios que disponen de un smartphone o laptop para conectarse a la Internet a través de una conexión inalámbrica. Sin pérdida de generalidad se asume que estos usuarios se mueven en un plano de dos dimensiones y no tienen ninguna restricción de movimiento. Mientras el usuario se mueve puede decidir enviar al LBS una consulta de rango respecto a su ubicación. Antes de hacerlo, el usuario debe escuchar el broadcast en curso y buscar si este contiene los elementos de datos (o puntos de interés) que responden a su consulta. Se asume que el equipo del usuario mantiene una cache local donde almacena la información obtenida del broadcast. Si el usuario no escucha toda la información requerida para responder su consulta (i.e no

recibe los POIs de las celdas que caen dentro de su consulta de rango), el mismo usuario bien puede construir una CUE con sus medios o pedir a un servidor de anonimato que la construya. Finalmente, el usuario transmite su CUE al LBS en reemplazo de la consulta de rango original.

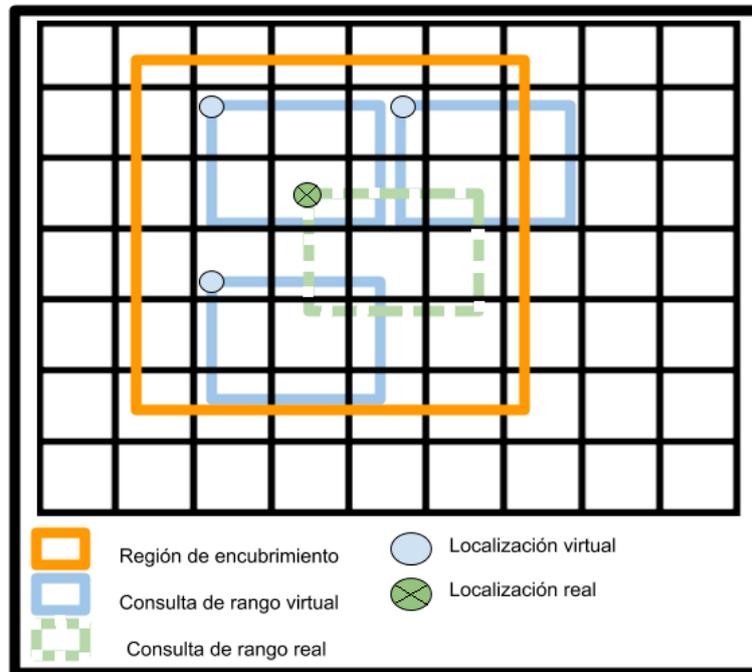
3. **Cola de consultas:** El LBS mantiene una cola de CUEs que esperan ser procesadas. Cuando el LBS culmina un broadcast de puntos de interés, éste observa esta cola de espera y procesa en batch todas estas consultas para luego ordenar sus respuestas mediante la ejecución de un algoritmo de scheduling. Basado en este ordenamiento, luego el LBS comienza a transmitir en un broadcast todas las respuestas a todos los usuarios del sistema. En nuestro trabajo se asume que el tiempo de procesamiento de las consultas es despreciable con respecto al tiempo que toma al broadcast difundir los elementos de datos.
4. **Broadcast:** Se modela la difusión o broadcast de los puntos de interés o datos mediante el empleo de una estructura de datos que representa la transmisión ordenada de las respuestas a las CUEs procesadas de la cola de consultas. A través del broadcast, el LBS transmite todos los elementos de datos solicitados por los usuarios, y en el peor de los casos puede resultar en la transmisión completa de todos los POIs de la base de datos.

### 3.1.1. Consultas con ubicación encubierta (CUEs)

En este trabajo se asume que se emplea la técnica del  $k$ -anonimato para anonimizar la posición de un usuario y, con ello, anonimizar su consulta de rango. Para anonimizar la posición, en la literatura se emplea el concepto de región de encubrimiento, el cual podemos definir como el conjunto de  $k$  posiciones altamente factibles de ser la posición real del usuario. Desde luego una de estas posiciones, es la real del usuario. Una forma de seleccionar estas posiciones factibles, consiste en dividir el área de servicio de los usuarios en una grilla. En la **Figura3.1** se muestra una grilla que consiste de 7 filas y 9 columnas y cada celda es de igual tamaño. Para anonimizar una consulta de rango se selecciona la celda que contiene la localización verdadera del usuario y otras  $k - 1$  celdas alternativas. Para cada posición alternativa se le asocia una consulta de rango de igual tamaño a la verdadera.

Entonces se define una *consulta con ubicación encubierta* como una consulta de rango que ha sido anonimizada con  $k$ -anonimato y que, por lo tanto, consiste de  $k$  consultas de rango, una de ellas es situada con respecto a la posición real del usuario y las otras corresponden a la misma consulta de rango pero situadas en las otras  $k - 1$  posiciones alternativas. Para facilitar la discusión acerca del procesamiento de una CUE, llamaremos

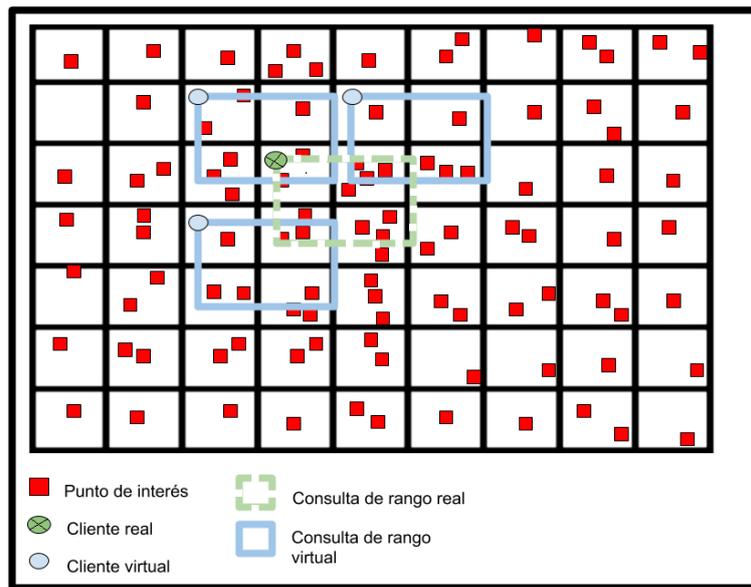
a las  $k$  consultas de rango que la componen, simplemente como las  $k$  subconsultas de una CUE.



**Figura 3.1:** Zona geográfica dividida en celdas, las cuales contienen usuarios en movimiento y subconsulta de rango.

A lo largo de este informe, emplearemos la siguiente notación. Sea  $q_1$  la consulta de rango de un usuario que luego de ser anonimizada se convierte en una CUE  $Q$ . Por definición, esta  $Q$  se compone de  $k$ -subconsultas de rango cuyo tamaño es igual a la  $q_1$  pero se encuentran cada una ubicadas en  $k - 1$  posiciones distintas

A continuación se presenta en la **Figura3.2** un ejemplo que contiene una CUE conformada por tres diferentes consultas de rango (con borde de color celeste) y en el interior de las las consultas de rango se observan los puntos de interés, cuyos detalles pueden ser enviados a los usuarios.



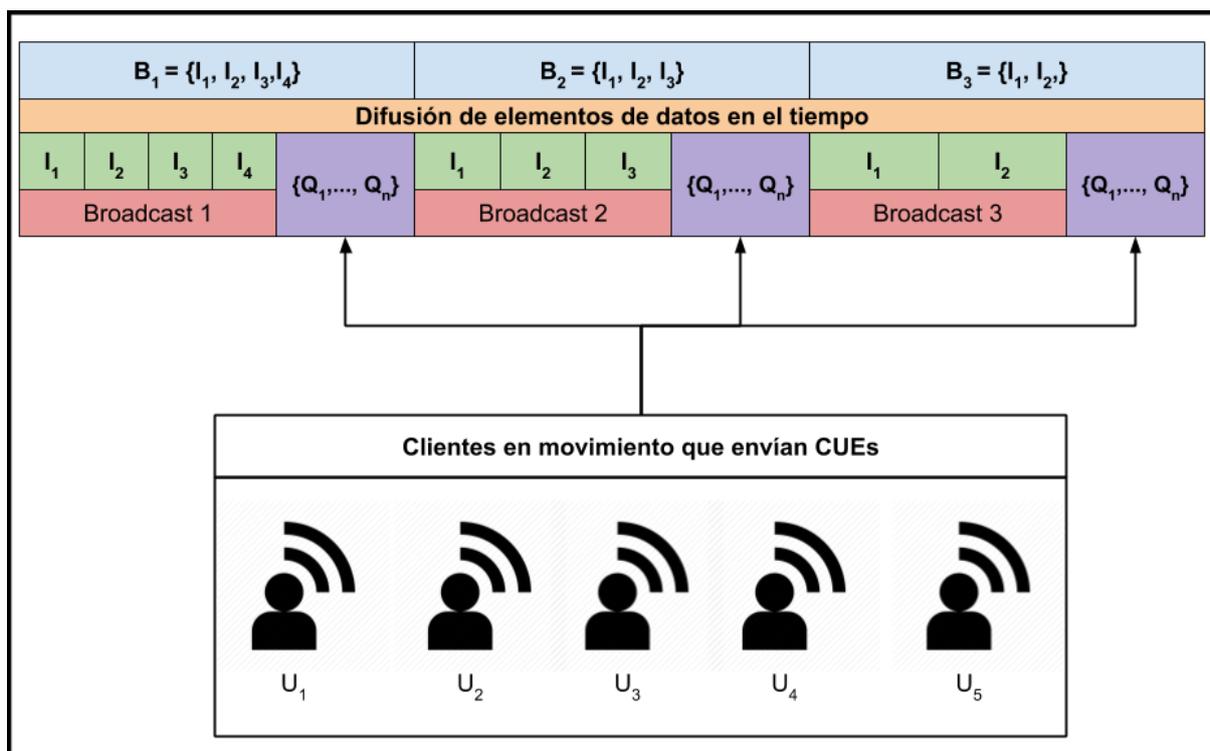
**Figura 3.2:** Zona geográfica dividida en celdas, las cuales contienen usuarios en movimiento, consultas de rango y puntos de interés.

Cuando el LBS recibe una CUE  $Q$ , este debe resolver cada una de las subconsultas que la componen. Resolver una subconsulta significa encontrar todos los Puntos de Interés (POIs) que se ubican dentro del rango. Por ejemplo, caracterizaremos cada POI por un conjunto fijo de atributos o ítems ( $I$ ) que poseen, por ejemplo, dirección, tipo de lugar, ofertas asociadas, etc. Sin perder la generalidad se asume en este trabajo que cada POI se caracteriza por un solo ítem ( $I$ ). Se considera que la respuesta a una subconsulta  $q$  se compone de todos los  $n$  POIs denotados como  $I_1, I_2, I_3 \dots I_n$ , cuya posición se encuentra dentro del rango espacial declarado por la subconsulta. Por lo tanto denotaremos la respuesta a una subconsulta  $q$  como  $q = \{I_1, I_2, I_3 \dots I_n\}$

Consideremos por ejemplo el caso presentado en la **Figura3.2**. Aquí se muestra una CUE etiquetada como  $Q$ , la cual se compone de cuatro subconsultas (rectángulos de color celeste) y los ítems en color rojo corresponden a los puntos de interés.

### 3.1.2. Ciclo de procesamiento y difusión de datos

En la **Figura 3.3** se muestra un grupo de usuarios denotados como  $U_1, U_2, U_3, U_4$  y  $U_5$  que envían sus CUEs al servidor LBS cada vez que sus consultas de rango no son respondidas por el broadcast en curso y también considere una secuencia de broadcasts denotados como  $B_1, B_2$  y  $B_3$ . Asuma que cuando el broadcast  $B_1$  es difundido entre estos usuarios, éstos comienzan a escuchar los elementos de datos que componen  $B_1$  y algunos de ellos concluyen al final de la transmisión que ésta no responde a sus consultas de rango. Sin perder la generalidad asuma que los usuarios  $U_1, U_2, U_3$  y  $U_4$  deciden enviar cada uno una CUEs  $Q_1, Q_2, Q_3$  y  $Q_4$  al LBS. Estas consultas son procesadas por el LBS y luego sus respuestas ordenadas y transmitidas en  $B_2$ . Mientras esto ocurre  $U_1, U_2$  y  $U_5$  crean nuevas consultas y, por lo tanto, buscan responderlas en el broadcast en curso. Del mismo modo señalado al final de  $B_1$ , aquellos usuarios que no son respondidos en el broadcast  $B_2$  enviarán sus consultas al LBS y éstas serán respondidas en el broadcast  $B_3$ . De este modo este ciclo de escuchar, enviar CUE, planificar y transmitir se vuelve a repetir.



**Figura 3.3:** Procedimiento en que el sistema envía los broadcast a los usuarios y cuando se termina de difundir un broadcast los usuarios quedan habilitados para enviar sus CUEs.

### 3.1.3. Broadcast

En este trabajo por simplicidad se asume que el servidor LBS puede también actuar como una estación base de una red inalámbrica y posee un único canal de difusión público por el cual se transmite datos periódicamente a todos los usuarios sin distinción. En algunos trabajos relacionados con el broadcasting de datos (Imielinski, Viswanathan, y Badrinath, 1997), también se incluye un índice aéreo que facilita a los usuarios conocer cuando la información deseada será escuchada. Esto no fue incluido en este trabajo y se pretende incorporar en un trabajo futuro.

De acuerdo a nuestro conocimiento, la mayoría de los trabajos que abordan el procesamiento de consultas con ubicación encubierta, proporcionan su respuesta directamente al usuario que la solicita. Este usuario deberá filtrar la información no deseada para obtener aquella que le será de utilidad. Para mitigar esta situación resulta ventajoso que las respuestas proporcionadas por el LBS sean difundidas a través de un canal de difusión abierto a todos los usuarios de modo que todos puedan sacar provecho de una respuesta.

El uso dado en la literatura al broadcasting de datos en redes inalámbricas asume implícitamente que todos estos datos son necesarios para alguien. Más aún, la difusión periódica de datos, permitiría a los usuarios proteger su privacidad de ubicación ya que primero buscarán dar respuesta a sus consultas en el broadcast en curso. Para aquellos que no encuentran respuesta en él, deberán encubrir su posición para acceder al LBS y luego esperar su respuesta en el siguiente broadcast. Con el fin de reducir el tiempo de respuesta de estos últimos usuarios, el LBS podría intentar determinar qué respuestas tienen mayor posibilidad de ser de utilidad a alguien y, de ese modo, otorgarles mayor prioridad al transmitir los datos en el broadcast.

En este trabajo se proponen 3 algoritmos de scheduling que permiten ordenar los elementos de datos que componen las respuestas y que luego son transmitidas en un broadcast: algoritmo basado en la envergadura, popularidad y relevancia. Estos algoritmos serán discutidos en un próximo capítulo.

Sólo para fines de extender este trabajo en el futuro, el simulador de broadcasting empleado en la sección experimental de este trabajo divide el broadcast en varios sub-broadcast de tamaño fijo. Esto se realizó, con el objeto de intentar alterar parcialmente una planificación de broadcast en curso. Sin embargo, en el presente trabajo, se decidió no realizar ninguna intervención ya que la modificación de un broadcast en curso podría permitir que algunas respuestas a ciertas CUEs pudieran experimentar de una latencia indefinida (problema de hambruna) y por lo tanto su uso fue declarado como trabajo futuro a investigar.

### 3.1.4. Métrica de Privacidad de Ubicación

En la literatura se encuentran diversas métricas para medir la privacidad, como las basadas en entropía (Serjantov y Danezis, 2002; Niu, Li, Zhu, Cao, y Li, 2015) y métricas basadas en distorsiones (Hoh y Gruteser, 2005). La primera es la más ampliamente usada en nuestro contexto y ha sido interpretada como la incertidumbre para determinar la ubicación real de un usuario entre un conjunto candidato de alternativas. En base a la entropía, se define una métrica de pérdida de la privacidad de ubicación cuando el usuario se ve obligado a enviar una consulta al LBS debido a que no encontró respuesta en el broadcast en curso

Para poder aplicar este concepto de entropía se requiere conocer con qué frecuencia son visitados los distintos lugares del área de servicio del sistema. En general, esta información hoy en día es considerada de dominio público y capturada por diversos servicios basados en la ubicación, como por ejemplo: Google Maps, Waze, Uber, solo por nombrar las más conocidas. Específicamente, dividimos el área de servicio en una grilla de  $N \times N$  celdas. Cada celda tiene una probabilidad de ser consultada (llamada probabilidad de consulta) que es proporcional al número de veces que se solicitó un servicio al LBS desde esa ubicación en particular en el pasado. Denotaremos como  $q_i$  a la probabilidad de consulta desde la celda  $i$ . Sabiendo que el área de servicio se divide en  $N^2$  celdas, luego se debe cumplir que  $\sum_{i=1}^{N^2} q_i = 1$ .

Considere ahora una región de encubrimiento, constituida por  $k$  ubicaciones (o celdas) contenidas en una CUE, de la cual sabemos que incluye una ubicación real y otras  $k - 1$  alternativas. Para cada celda o ubicación se define una probabilidad condicional de ser la ubicación real. Sea  $p_i$  ( $i = 1, 2, \dots, k$ ) la probabilidad de que la  $i$ -ésima ubicación alternativa sea la real y ésta se define como  $p_i = q_i / \sum_{i=1}^k q_i$ .

Entonces se define la entropía  $H$  para identificar la ubicación real en una región de encubrimiento de tamaño  $k$  de una CUE ( $Q$ ) como:

$$H(Q) = \sum_{i=1}^k -p_i \cdot \log_2 p_i \quad (3.1)$$

Esta noción de entropía establece que mientras mayor sea su valor, mayor será la incertidumbre de descubrir la posición real (y menor la pérdida de privacidad), y ésta alcanza su valor máximo cuando todas las probabilidades condicionales son iguales y en este caso resulta ser igual a  $\log_2 k$ .

Por lo tanto, se define la pérdida de privacidad de ubicación para un usuario que decide

enviar una CUE ( $Q$ ) al LBS como  $Per(Q) = \frac{1}{H(Q)}$ . Note que esta métrica no considera la protección a la privacidad que se logra cuando se encuentra respuesta a una consulta en el broadcast. Para capturar este efecto extendemos nuestra noción de privacidad del siguiente modo.

Para una consulta de rango respondida en el broadcast, el servidor LBS no obtiene información sobre la ubicación real del usuario y, por lo tanto, cada ubicación (o celda para ser más preciso) del área de aplicación puede ser la ubicación real. Entonces al aplicar la ecuación 3.1, la entropía de identificar la ubicación del usuario al área de servicio es  $\log_2 N^2$ . Entonces se define la pérdida de privacidad de ubicación cuando un usuario encuentra en primera instancia respuesta a su consulta en el broadcast como  $Per_{Broadcast} = \frac{1}{\log_2 N^2}$ . Note que esta definición cumple lo siguiente  $Per(Q) \geq \frac{1}{\log_2 k} \geq Per_{Broadcast}$ .

Habiendo presentado lo anterior, podemos ahora definir una métrica de pérdida de privacidad para un usuario  $U$  al usar nuestro sistema. Sea  $U$  un usuario móvil que durante un cierto periodo de tiempo realiza una serie de consultas dependientes de su ubicación al sistema propuesto. Del total de consultas,  $N_{LBS}$  CUEs (identificadas como  $(Q_i)$ ) son enviadas al LBS y  $N_{Broadcast}$  son respondidas inmediatamente en el broadcast en curso. De este modo se define la pérdida de privacidad para este usuario ( $Per(U)$ ) como:

$$Per(U) = \sum_{i=1}^{N_{LBS}} Per(Q_i) + N_{Broadcast} \cdot Per_{Broadcast} \quad (3.2)$$

Note que para mejorar la privacidad se puede ya sea aumentar el grado de anonimato  $k$  de una CUE o bien intentar aumentar el valor de  $N_{Broadcast}$ .

Finalmente, se define una noción de privacidad global para el sistema completo. Sea  $N_U$  el número total de usuarios ( $U_i$ ) accediendo a nuestro sistema durante un cierto periodo de tiempo. Entonces la pérdida de privacidad (promedio) de nuestro sistema se define como:

$$Per(Sistema) = \frac{1}{N_U} \sum_{i=1}^{N_U} Per(U_i) \quad (3.3)$$

# Capítulo 4

## Estrategias de scheduling propuestas

En esta investigación se busca comprobar la viabilidad de diversas técnicas scheduling para ordenar las respuestas a un conjunto de CUEs de modo tal de mejorar en la medida de lo posible el tiempo de respuesta, la obsolescencia, la utilidad, la efectividad, la justicia, la carga de trabajo y la privacidad. Uno de los principales aportes de esta investigación son los algoritmos de scheduling que ordenan las respuestas de CUEs que están encoladas y por lo tanto en espera de ser procesadas. También se consideran aspectos como la transmisión de respuestas en tiempo real, el movimiento de los usuarios en un plano, la dependencia de una CUE con múltiples elementos de datos y la incertidumbre del servidor respecto a los elementos de datos que serán de utilidad para los usuarios.

En la literatura se pueden observar diversas técnicas de planificación. Las cuales apuntan a mejorar, de forma individual, diversas métricas de evaluación del rendimiento de un broadcast. En esta investigación, los algoritmos de scheduling influyen en más de una métricas de evaluación vistas en el **capítulo 5**.

### 4.1. Antecedentes previos

En este capítulo presentamos la operación de un LBS consciente de la ubicación de sus usuarios. Primero, este LBS recibe un conjunto de CUEs y cada una de ellas es procesada y su respuesta determinada. Luego, todas las respuestas son ordenadas por un algoritmo de scheduling. Finalmente, el LBS transmite las respuestas ordenadas en un broadcast público a todos los usuarios inalámbricos que se mueven en un área geográfica. Por lo tanto, desde ahora este trabajo se centra en la presentación de los algoritmos de scheduling desarrollados.

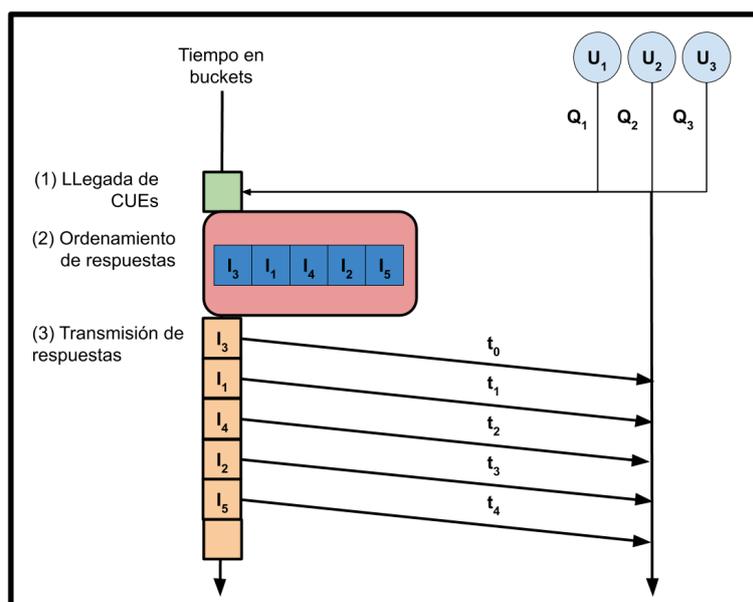
Brevemente describiremos cómo una CUE enviada por un usuario al servidor LBS es procesada. Siguiendo la notación presentada en el **Capítulo 3**, considere que una CUE se

descompone en un serie de subconsultas:

- $Q \equiv \{q_1, q_2, q_3, q_k\}$

El servidor LBS descompone cada CUE en sus correspondientes subconsultas. Por lo tanto se define que si  $q$  es una subconsulta de una CUE  $Q$ , entonces, por transitividad, un elemento de dato  $I$  que responde una subconsulta  $q$  también responde a  $Q$ . Dado lo anterior, considere el siguiente ejemplo:

- Una CUE que se compone de las siguientes subconsultas :  $Q_1 \equiv \{q_{1,1}, q_{2,1}, q_{3,1}, q_{4,1}\}$
- Si la respuesta a  $q_{1,1}$  es  $\{I_1, I_2, I_3\}$ , entonces los tres elemento de datos también deben ser parte de la solución a  $Q_1$



**Figura 4.1:** Representación del proceso de llegada de las CUEs al servidor LBS (1), Ordenamiento de los elementos de datos que responden las CUEs (2) y la transmisión de las respuestas para todos los usuarios del sistema(3). Se considera que el tiempo transcurrido es contado en buckets. Por otra parte, se condera que los elementos de datos o respuestas transmitidas(3) son recibidos por los usuarios con un leve desfase de tiempo.

Considere la **Figura4.1** en la cual se observa una representación del proceso de envío de consultas, ordenamiento de respuestas y la difusión o broadcasting de las respuestas. Considere tres usuarios  $u_1, u_2$  y  $u_3$  los cuales crean y envían sus CUEs  $Q_1, Q_3, Q$  respectivamente al servidor LBS. Luego, las CUEs son procesadas y se determina que los elementos de

datos  $I_1, I_2, I_3, I_4, I_5$  son requeridos para responder las tres CUES. Por otro lado, los algoritmos de scheduling ordenan estas respuestas, por ejemplo:  $\{I_3, I_1, I_4, I_2, I_5\}$  y luego difundidas públicamente de forma secuencial a los tres usuarios. Es decir,  $I_3$  es transmitido en  $t_0$ , luego  $I_1$  en  $t_2$ , y así sucesivamente.

Los algoritmos de scheduling propuesto en este trabajo, se pueden expresar del siguiente modo:

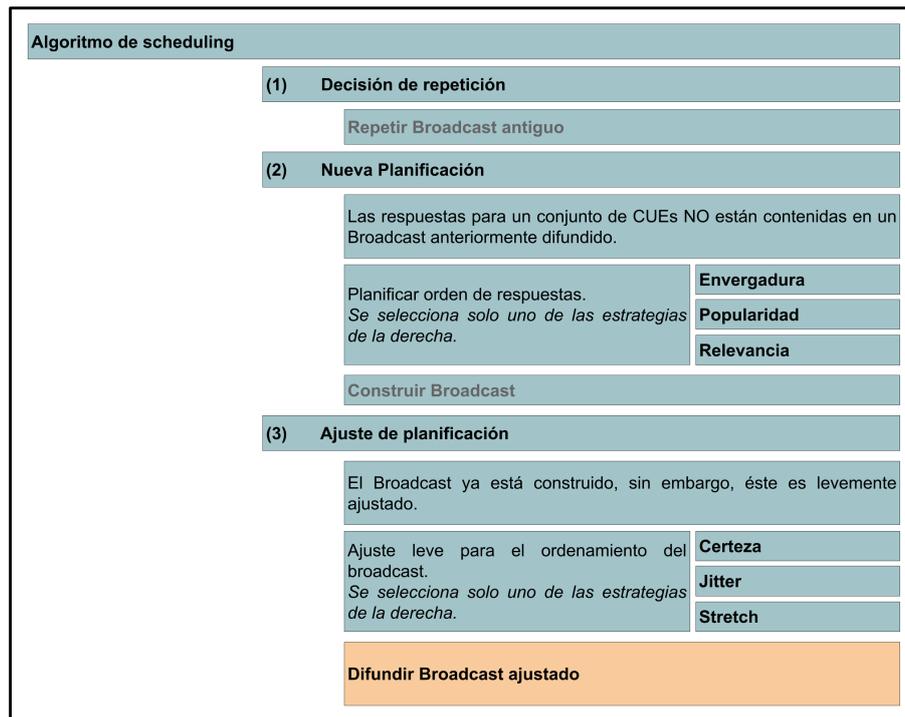


Figura 4.2: Procedimiento para la planificación del broadcast.

**Fase de decisión de repetición (1):** El objetivo de esta etapa es decidir repetir un broadcast. Por simplicidad hemos establecido un número fijo de repeticiones de broadcast y como trabajo futuro se buscará determinar predictores adecuados para le señalen al servidor LBS cuando es conveniente o no de una repetición.

**Fase de planificación (2):** a partir de un algoritmo de scheduling, se genera un ordenamiento de las respuestas que se envían en un broadcast. A diferencia de otros trabajos, en esta fase cuando se planifican las CUES de los usuarios, en el nuevo broadcast estarán los elementos encontrados en la anterior difusión de elementos y los que no fueron encontrados, luego, el nuevo broadcast es concatenado como una parte de la anterior difusión, pero de la anterior difusión se eliminan los elementos que están en el nuevo broadcast, de modo que no existan elementos repetidos en la nueva transmisión del broadcast ajustado.

**Fase de ajuste de planificación (3):** Se aplica un leve ajuste al ordenamiento definido por la etapa anterior, con el fin de observar si el ordenamiento propuesto puede ser mejorado con respecto a otras métricas de evaluación empleadas por el servidor LBS.

## 4.2. Algoritmo de scheduling basado en la envergadura

La técnica de scheduling de *envergadura* (E) tiene como objetivo responder tantas consulta como sea posible en el menor lapso de tiempo, es decir, responder el mayor número de CUEs con el menor número de buckets. Ésta se enfoca en mejorar la carga de trabajo del servidor y por esto se propone que las CUEs con menor tamaño deben tener mayor prioridad en el broadcast, de modo que si se difunden primero las respuestas de CUEs pequeñas entonces en menor tiempo se responden más CUEs. Por ejemplo, sea  $C'$  una cola de espera que contiene dos CUEs  $C' \equiv \{Q_1, Q_2\}$ , de las cuales se sabe lo siguiente:

- $Q_1 \equiv \{q_1, q_2, q_3, q_4\}$ 
  - $q_1 \Rightarrow \{I_1, I_2, I_3\}$
  - $q_2 \Rightarrow \{I_3, I_4\}$
  - $q_3 \Rightarrow \{I_5, I_6, I_7, I_8\}$
  - $q_4 \Rightarrow \{I_9, I_{10}, I_{11}, I_{12}\}$
- $Q_2 \equiv \{q_1, q_2, q_3\}$ 
  - $q_1 \Rightarrow \{I_{56}, I_{57}, I_{58}\}$
  - $q_2 \Rightarrow \{I_{58}, I_{59}, I_{60}\}$
  - $q_3 \Rightarrow \{I_{60}, I_{61}, I_{62}, I_{63}\}$

Dado lo anterior planteado, por transitividad se observa que las CUEs  $Q_1$  y  $Q_2$  requieren los siguientes elementos de datos:

- $Q_1 \equiv \{I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11}, I_{12}\}$
- $Q_2 \equiv \{I_{56}, I_{57}, I_{58}, I_{59}, I_{60}, I_{61}, I_{62}, I_{63}\}$

La envergadura es una métrica que evalúa el número de elementos de datos que compone una CUE. La estrategia de ordenamiento de respuestas que utiliza la envergadura priorizar

las respuestas que pertenecen a CUEs que requieren una baja cantidad de elementos, se considera la siguiente **ecuación 4.1** para calcular la envergadura de una CUE:

$$E(Q_i) = \frac{1}{|Q_i|} \quad (4.1)$$

Considerando el valor de **envergadura** de las CUEs como  $|Q_1| = 12$  y  $|Q_2| = 8$ . Entonces, según la **ecuación 4.1** se puede concluir que  $E(Q_1) < E(Q_2)$  esto significa que la CUE  $Q_2$  tiene mayor prioridad en el broadcasting. En la **ecuación 4.1** se entiende  $|Q_i|$  como el número de elementos de datos requeridos por una CUE  $Q_i$ , entonces su inverso multiplicativo permite que las CUEs con menor número de elementos de datos requeridos tenga mayor valor de envergadura y de este modo las CUEs con mayor  $E$  tienen mayor prioridad en el broadcasting.

Un algoritmo que emplea la técnica de envergadura para ordenar los elementos de datos de un broadcasting busca la CUE  $Q_i$  que tiene mayor envergadura y agrupa los elementos de datos en el mismo orden que son solicitados. Por ejemplo, sabiendo que  $E(Q_1) = \frac{1}{12} = 0,083$  y  $E(Q_2) = \frac{1}{8} = 0,125$ , los elementos de datos de la CUE  $Q_2$  deben ser agrupados primero en el broadcasting. Por lo tanto, un broadcasting  $B$  se compone por los elementos de datos de las CUEs  $B = \{Q_2 \cup Q_1\}$ , entonces:

- $B = \{I_{56}, I_{57}, I_{58}, I_{59}, I_{60}, I_{61}, I_{62}, I_{63}, I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11}, I_{12}\}$

La operación de unión ( $\cup$ ) indica que los elementos de datos de un broadcasting no están repetidos en el caso en que se encuentran más de una CUE. A modo de simplificar la posición de un elemento de dato repetido, se conserva la posición del elemento de datos que pertenece a la CUE con mayor prioridad. Por ejemplo, dada las siguientes CUEs:

- $Q_3 = \{I_1, I_2, I_3, I_4\}$

- $Q_4 = \{I_2, I_5, I_6, I_7, I_8\}$

Se obtiene que  $E(Q_3) = \frac{1}{4} = 0,25$  y  $E(Q_4) = \frac{1}{5} = 0,2$ , por lo tanto el broadcasting  $B = \{Q_3 \cup Q_4\}$ , donde el elemento de dato  $I_2$  solicitado por ambas CUEs es ubicado de la siguiente forma:

- $B = \{I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8\}$

A continuación se presenta el algoritmo de scheduling por envergadura (E):

---

**Algoritmo 1:** ALGORITMO DE ENVERGADURA(E)

---

**Entrada:** Cola de consultas  
**Salida:** Orden en que las respuestas a las CUEs serán transmitidas

- 1 R = arreglo de CUEs ponderadas;
- 2 B = Conjunto de broadcasting;
- 3 **para**  $T = 1; T < \text{Repeticiones}; T++$  **hacer**
- 4 | Transmisión de broadcasting anterior;
- 5 **fin**
- 6 **para**  $i=1; i < \text{Numero de CUEs}; i++$  **hacer**
- 7 | Para cada CUEs  $Q_i$  calcular envergadura: **ecuación 4.1**;
- 8 |  $R[i] = E(Q[i]);$
- 9 **fin**
- 10 **para**  $H=1; H < \text{Broadcast almacenados}; H++$  **hacer**
- 11 | **mientras**  $R \neq \emptyset$  **hacer**
- 12 | | E = Seleccionar desde R con probabilidad **ecuación 4.2** los elementos de datos de una CUE;
- 13 | | b = agregar E a la cola;
- 14 | **fin**
- 15 | B = guardar broadcast b;
- 16 **fin**
- 17  $\text{Broadcast} = \text{Seleccionar b de B según: certeza, stretch o Jitter};$
- 18 **devolver** *Broadcast*

---

En la línea 3 del **algoritmo 1** se especifica que antes de entrar en el algoritmo de envergadura, un broadcasting fue transmitido previamente una determinada cantidad de veces. Esta repetición es realizada porque se asume que los clientes van cambiando de posición (y por ende de elementos de datos requeridos) de forma gradual.

En la línea 6 se calcula el nivel de envergadura de todas las CUEs que arriban al servidor LBS. Este valor está dado por la **ecuación 4.1**.

En la línea 10 se utiliza una selección probabilista de las respuestas de las CUEs. Entonces, la probabilidad  $PE_{Q_i}$  de una CUE se normaliza del siguiente modo:

$$PE(Q_i) = \frac{E(Q_i)}{\sum_{j=1}^N E(Q_j)} \quad (4.2)$$

Donde  $PE(Q_i)$  está dada por la **ecuación 4.2**,  $N$  es el número de CUEs que se evalúan de

una cola de espera  $C' = \{Q_1, Q_2, Q_3, \dots, Q_N\}$  y  $\sum_{j=1}^N E(Q_j)$  es la suma de todas las probabilidades de las CUEs de una cola de espera. Por ejemplo, dadas las anteriores CUEs  $Q_3$  y  $Q_4$ , se obtiene que:

- $PE(Q_3) = \frac{0,25}{0,45} = 0,55$
- $PE(Q_4) = \frac{0,2}{0,45} = 0,45$

Según los anteriores datos de las CUEs, un broadcasting ordenado con la técnica **E** indica que la CUE  $Q_3$  tiene un 55% de posibilidades de estar en la primera posición del broadcasting, mientras que la CUE  $Q_4$  tiene un 45% de posibilidades de estar en el primer lugar del broadcasting.

En la línea 17 del **algoritmo 1** se aplica una selección de los broadcast acumulado, esta selección se explica con detalle en la **sección 4.5**. Este último será llamado broadcast ajustado ya sea por certeza jitter o stretch.

Por otro lado, se presenta la técnica **ExW** la cual agrega al valor de envergadura el tiempo de espera. De este modo la probabilidad de envergadura  $PE(Q_i)$  con el tiempo de espera  $W(Q_i)$  son considerados un nuevo factor ponderador. Entonces, la probabilidad para la técnica **ExW** se calcula como ponderación entre la probabilidad **PE** y el tiempo de espera  $W$ ; ponderación que se normaliza. Por ejemplo, una cola de espera  $C'$  tiene 3 CUEs  $Q' = \{Q_5, Q_6, Q_7\}$ , de las cuales se sabe lo siguiente:

- $Q_5 = \{I_1, I_2, I_3, I_4\}$
- $Q_6 = \{I_1, I_5, I_6, I_7, I_8\}$
- $Q_7 = \{I_5, I_9, I_{10}, I_{11}, I_{12}, I_{13}\}$

Entonces se obtiene que  $E(Q_5) = 0,25$ ,  $E(Q_6) = 0,2$  y  $E(Q_7) = 0,17$ , de este modo se concluye que  $\sum_{j=1}^N E(Q_j) = 0,62$ , por lo tanto:

- $PE(Q_5) = 0,40$
- $PE(Q_6) = 0,32$
- $PE(Q_7) = 0,28$

Para calcular la probabilidad para la técnica **ExW** se requiere el tiempo de espera  $W$  de las CUEs, de este modo se tiene los siguientes datos:

- $W(Q_5) = 8$
- $W(Q_6) = 10$
- $W(Q_7) = 12$

Donde  $W(Q_i)$  corresponde al tiempo de espera de una CUE medidos en buckets. Sin perder la generalidad, al ponderar  $PE(Q_i) \cdot W(Q_i)$  la suma de las probabilidades es distinto de 1. Por ello esta ponderación se debe normalizar, de modo que el algoritmo de ordenamiento probabilista trabaje con valores que se pueda interpretar, es por ello que la probabilidad **PExW** de una CUE es normalizada del siguiente modo:

$$PExW(Q_i) = \frac{PE(Q_i) \cdot W(Q_i)}{\sum_{j=1}^N PE(Q_j) \cdot W(Q_j)} \quad (4.3)$$

Donde  $PE(Q_i)$  esta dada por la **ecuación 4.2**,  $N$  es el número de CUEs que se evalúan de una cola de espera  $Q' = \{Q_1, Q_2, Q_3, \dots, Q_N\}$  y  $\sum_{j=1}^N PE(Q_j) \cdot W(Q_j)$  es la suma de todas las probabilidades  $EP(Q_j)$  de las CUEs de una cola de espera con su respectivo tiempo de espera  $W(Q_j)$ . Por ejemplo, dadas las anteriores CUEs  $Q_5$ ,  $Q_6$  y  $Q_7$ , según la **ecuación 4.3** se obtiene que:

- $PExW(Q_5) = \frac{3,2}{9,76} = 0,33$
- $PExW(Q_6) = \frac{3,2}{9,76} = 0,33$
- $PExW(Q_7) = \frac{3,36}{9,76} = 0,34$

Según los anteriores datos de las CUEs, un broadcasting ordenado con la ponderación **PExW** indica que la CUE  $Q_5$  tiene un 33% de posibilidades de estar en la primera posición del broadcasting, la CUE  $Q_6$  tiene un 33% de posibilidades de estar en la primera posición del broadcasting y la CUE  $Q_7$  tiene un 34% de posibilidades de estar en el primer lugar del broadcasting. Por ejemplo, un algoritmo probabilista puede entregar el siguiente broadcasting:

- $B = \{I_1, I_2, I_3, I_4, I_5, I_9, I_{10}, I_{11}, I_{12}, I_{13}, I_6, I_7, I_8\}$

A continuación se presenta el algoritmo de scheduling por envergadura con tiempo de

espera (ExW):

---

**Algoritmo 2:** ALGORITMO DE ENVERGADURA CON TIEMPO DE ESPERA (EXW)

---

**Entrada:** Cola de consultas

**Salida:** Orden en que las respuestas a las CUEs serán transmitidas

```

1 R = arreglo de CUEs ponderadas;
2 B = Conjunto de broadcasting;
3 para  $T = 1; T < Repeticiones; T++$  hacer
4   | Transmisión de broadcasting anterior;
5 fin
6 para  $i = 1; i < Numero\ de\ CUEs; i++$  hacer
7   | Para cada CUEs  $Q_i$  calcular envergadura: ecuación 4.2;
8   |  $R[i] = PE(Q[i]);$ 
9 fin
10 para  $H = 1; H < Broadcast\ almacenados; H++$  hacer
11   | mientras  $R \neq \emptyset$  hacer
12     | E = Seleccionar desde R con probabilidad ecuación 4.3 los elementos de datos
13     | de una CUE;
14     | b = agregar E a la cola;
15   | fin
16   | B = guardar broadcast b;
17 fin
18 Broadcast = Seleccionar b de B según: certeza, stretch o Jitter;
19 devolver Broadcast

```

---

En el **algoritmo 2** se cuenta con los mismos pasos e instrucciones que el **algoritmo 1**, sin embargo estos difieren en la línea 8 y 12, donde en el **algoritmo 2** se agrega el tiempo de espera a la ponderación de la probabilidad para el algoritmos de ordenamiento por selección.

### 4.3. Algoritmo basado en la popularidad

La técnica de scheduling de **popularidad(P)** tiene como objetivo responder tan pronto como sea posible las CUEs más requeridas, es decir, cuando una gran cantidad de usuarios que solicitan elementos de datos al servidor LBS se concentra en una determinada zona, se priorizan sus respuestas. El hecho de medir la popularidad de los elementos de datos se propone por el fenómeno de la redundancia espacio temporal. Este fenómeno implica que los usuarios que se concentran en una determinada zona del mapa, se trasladan de forma

paulatina hasta otros sectores. Por ejemplo, sea  $C_r$  una cola de espera que contiene 3 CUEs  $C_r \equiv \{Q_1, Q_2, Q_3\}$  de las cuales se sabe lo siguiente:

- $Q_1 \equiv \{I_1, I_2, I_3, I_4, I_5\}$
- $Q_2 \equiv \{I_4, I_6, I_7\}$
- $Q_3 \equiv \{I_4, I_6, I_8, I_9\}$

EL valor de **popularidad de un elemento de dato** esta dado por la **ecuación 4.4**, donde a partir de un elemento de dato se debe buscar el número de CUEs que lo solicitan.

$$P(I_j) = \|I_j \in Q_r\| \quad (4.4)$$

Según los anteriores datos entregados se puede observar los siguientes valores de popularidad para los elementos de datos:

- $I_1 \in \{Q_1\} \implies P(I) = 1$
- $I_2 \in \{Q_1\} \implies P(I) = 1$
- $I_3 \in \{Q_1\} \implies P(I) = 1$
- $I_4 \in \{Q_1, Q_2, Q_3\} \implies P(I) = 3$
- $I_5 \in \{Q_1\} \implies P(I) = 1$
- $I_6 \in \{Q_2, Q_3\} \implies P(I) = 2$
- $I_7 \in \{Q_2\} \implies P(I) = 1$
- $I_8 \in \{Q_3\} \implies P(I) = 1$
- $I_9 \in \{Q_3\} \implies P(I) = 1$

Donde un elemento de dato  $I_1$  esta contenido por  $v > 0$  CUEs  $I_i \in \{Q_1, \dots, Q_v\}$ . Se observa que el elemento de dato más solicitado es  $I_4$ , de este modo debe tener mayor prioridad en el broadcasting. Por lo tanto, un broadcasting  $B$  se compone por los elementos de datos ordenados desde mayor a menor popularidad de la siguiente forma:

- $B = \{I_4, I_6, I_1, I_2, I_3, I_5, I_7, I_8, I_9\}$

Al igual que la técnica de envergadura, en la técnica de popularidad no se repiten elementos de datos en el broadcasting. En ésta técnica es más sencillo de ver el hecho de no repetir los elementos de datos, dado que para cada uno de ellos se calcula la popularidad y según su valor son puestos en el broadcasting. Por otro lado, cuando los elementos de datos tienen igual popularidad, se acomodan primero los elementos de datos que pertenecen a las CUEs que están primero en la cola de consultas. Dicho de otro modo, dado que los elementos de datos  $I_1, I_2, I_3$ , pertenecen a la CUE  $Q_1$ , entonces deben estar primero que  $I_5, I_7, I_8, I_9$ .

A continuación se presenta el algoritmo de scheduling de popularidad (P):

---

**Algoritmo 3:** ALGORITMO DE POPULARIDAD(P)

---

**Entrada:** respuestas requeridas por los usuarios

**Salida:** Orden en que las respuestas a las CUEs serán transmitidas

```

1 R = arreglo de elementos de datos ponderados;
2 B = Conjunto de broadcasting;
3 para  $T = 1; T < \text{Repeticiones}; T++$  hacer
4   | Transmisión de broadcasting anterior;
5 fin
6 para  $i=1; i < \text{Numero de elementos de datos}; i++$  hacer
7   | Para cada Elemento de dato  $I_i$  calcular popularidad: ecuación 4.4;
8   |  $R[i] = P(I[i]);$ 
9 fin
10 para  $H=1; H < \text{Broadcast almacenados}; H++$  hacer
11   | mientras  $R \neq \emptyset$  hacer
12     | E = Seleccionar desde R con probabilidad ecuación 4.5 un elemento de dato;
13     | b = agregar E a la cola;
14   | fin
15   | B = guardar broadcast b;
16 fin
17  $\text{Broadcast} =$  Seleccionar b de B según: certeza, stretch o Jitter;
18 devolver Broadcast
```

---

Como conclusión, la técnica de popularidad presentada en el **algoritmo 3** tiene la misma forma que la técnica de envergadura, con la diferencia de que con la técnica de popularidad se busca el nivel de requerimiento de los elementos de datos, de modo que ellos toman el protagonismo en el ordenamiento del broadcasting. Con la técnica de popularidad es posible que se generen broadcasting que en las primeras posiciones tengan elementos de datos de varias CUEs que requieren diferentes elementos de datos.

La técnica **P** es utilizada por un algoritmo de ordenamiento probabilista. De modo que el ponderador  $PP(I_i)$  es considerado una probabilidad. Al igual que la técnica de envergadura, la popularidad debe ser normalizada de modo que las probabilidades de cada uno de los elementos de dato sume 1. Es por ello que el valor  $P(I_i)$  de un elemento de dato es normalizada de la siguiente forma:

$$PP(I_j) = \frac{P(I_i)}{\sum_{j=1}^N P(I_j)} \quad (4.5)$$

Donde  $P(I_i)$  es dado por la **ecuación 4.4**,  $N$  es el número de CUEs que contienen al elemento de dato  $I_j$  y  $\sum_{j=1}^N P(I_j)$  es la suma de todas las probabilidad de los elementos de datos requeridos por las CUEs de una cola de consultas. Por ejemplo, dadas las CUEs  $Q' = \{Q_1, Q_2, Q_3\}$  se obtiene que :

- $PP(I_1) = \frac{1}{12} = 0,08$
- $PP(I_2) = \frac{1}{12} = 0,08$
- $PP(I_3) = \frac{1}{12} = 0,08$
- $PP(I_4) = \frac{3}{12} = 0,27$
- $PP(I_5) = \frac{1}{12} = 0,08$
- $PP(I_6) = \frac{2}{12} = 0,17$
- $PP(I_7) = \frac{1}{12} = 0,08$
- $PP(I_8) = \frac{1}{12} = 0,08$
- $PP(I_9) = \frac{1}{12} = 0,08$

Según los anteriores datos de los elementos, un broadcasting ordenado con la técnica **PP** indica que los elementos de datos  $\{I_1, I_2, I_3, I_5, I_7, I_8, I_9\}$  tienen un 8% de posibilidades de estar en las primeras posiciones, el elemento de dato  $I_6$  tiene 17% de posibilidades de estar en las primeras posiciones del broadcasting y el elemento de dato  $I_4$  tiene una probabilidad del 27% de estar en las primeras posiciones del broadcasting.

Por otro lado, se presenta la técnica **PxW** la cual agrega al valor de popularidad el tiempo de espera del elemento de dato. Para ello se selecciona el mayor tiempo de espera de las CUEs que requieren un elemento de dato. De este modo se obtienen los siguientes datos para las CUES  $Q_1, Q_2, Q_3$ :

- $W(Q_1) = 8$
- $W(Q_2) = 10$
- $W(Q_3) = 12$

Entonces, dados los anteriores datos de tiempo de espera, los elementos de datos contiene los siguientes tiempo de espera:

- $W(I_1) = 8$
- $W(I_2) = 8$
- $W(I_3) = 8$
- $W(I_4) = 12$
- $W(I_5) = 8$
- $W(I_6) = 12$
- $W(I_7) = 10$
- $W(I_8) = 12$
- $W(I_9) = 12$

El múltiplo de las ponderaciones  $PP(I_i) \cdot W(I_i)$  como probabilidad para un algoritmo de ordenamiento probabilista es distinto de 1 al combinarlo con varios elementos de datos. Es por ello que para calcular  $PP \times W$  se requiere normalizar el múltiplo entre la ponderación probabilista de la popularidad y el tiempo de espera del elemento de dato. De este modo se obtiene la siguiente probabilidad:

$$PP \times W(I_j) = \frac{PP(I_i) \cdot W(I_i)}{\sum_{j=1}^N PP(I_j) \cdot W(I_j)} \quad (4.6)$$

Donde  $PP(I_i)$  esta dada por la **ecuación 4.5**,  $N$  es el número de CUEs que contienen al elemento de dato  $I_j$  y  $\sum_{j=1}^N PP(I_j) \cdot W(I_j)$  es la suma de todas las probabilidades  $PP(Q_i)$  de las CUEs de una cola de espera con su respectivo tiempo de espera  $W(I_j)$ . Por ejemplo, dadas los anteriores elementos de datos  $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9$ , según la **ecuación 4.6** se obtiene que:

- $PPxW(I_1) = \frac{0,08 \cdot 8}{10,56} = 0,06$
- $PPxW(I_2) = \frac{0,08 \cdot 8}{10,56} = 0,06$
- $PPxW(I_3) = \frac{0,08 \cdot 8}{10,56} = 0,06$
- $PPxW(I_4) = \frac{0,27 \cdot 12}{10,56} = 0,31$
- $PPxW(I_5) = \frac{0,08 \cdot 8}{10,56} = 0,06$
- $PPxW(I_6) = \frac{0,17 \cdot 12}{10,56} = 0,19$
- $PPxW(I_7) = \frac{0,08 \cdot 10}{10,56} = 0,08$
- $PPxW(I_8) = \frac{0,08 \cdot 12}{10,56} = 0,09$
- $PPxW(I_9) = \frac{0,08 \cdot 12}{10,56} = 0,09$

Según los anteriores datos de los elementos, un broadcasting ordenado con la técnica **PP-W** indica que los elementos de datos  $\{I_1, I_2, I_3, I_5, I_7\}$  tienen un 8% de posibilidades de estar en las primeras posiciones, el elemento de dato  $I_4$  tiene 31% de posibilidades de estar en las primeras posiciones del broadcasting, el elemento de dato  $I_6$  tiene 19% de posibilidades de estar en las primeras posiciones del broadcasting y los elemento de datos  $I_8, I_9$  tiene una probabilidad del 9% de estar en las primeras posiciones del broadcasting.

A continuación se se presenta el algoritmo de scheduling de popularidad con tiempo de

espera(PxW):

---

**Algoritmo 4:** ALGORITMO DE POPULARIDAD CON TIEMPO DE ESPERA(PxW)

---

**Entrada:** respuestas requeridas por los usuarios

**Salida:** Orden en que las respuestas a las CUEs serán transmitidas

```

1 R = arreglo de elementos de datos ponderados;
2 B = Conjunto de broadcasting;
3 para  $T = 1; T < Repeticiones; T++$  hacer
4   | Transmisión de broadcasting anterior;
5 fin
6 para  $i=1; i < Numero\ de\ elementos\ de\ datos; i++$  hacer
7   | Para cada Elemento de dato  $I_i$  calcular popularidad: ecuación 4.5;
8   |  $R[i] = P(I[i]);$ 
9 fin
10 para  $H=1; H < Broadcast\ almacenados; H++$  hacer
11   | mientras  $R \neq \emptyset$  hacer
12     | E = Seleccionar desde R con probabilidad ecuación 4.6 un elemento de dato;
13     | b = agregar E a la cola;
14   | fin
15   | B = guardar broadcast b;
16 fin
17 Broadcast = Seleccionar b de B según: certeza, stretch o Jitter;
18 devolver Broadcast

```

---

El **algoritmo 4** difiere del **algoritmo 3** en la línea 3. Donde se utiliza como ponderador el producto normalizado de la probabilidad de popularidad  $PP(I_i)$  y el tiempo de espera  $W(I_i)$ .

#### 4.4. Algoritmo basado en la relevancia

La técnica de scheduling de **relevancia (R)** tiene como objetivo responder tan pronto como sea posible las CUEs más requeridas con el menor índice de privacidad, es decir, cuando una gran cantidad de usuarios que solicitan elementos de datos al servidor LBS se concentra en una determinada zona y estos deciden ocultar su ubicación con un bajo nivel de anonimato, se priorizan sus respuestas. El hecho de medir la relevancia de los elementos de datos se propone por el fenómeno de la redundancia espacio temporal y las decisiones de privacidad de los usuarios. Por ejemplo, sea  $C'$  una cola de espera que contiene 3 CUEs  $C' \equiv \{Q_1, Q_2\}$  de las cuales se sabe lo siguiente:

- $Q_1 \equiv \{q_1, q_2, q_3\}$ 
  - $q_1 \Rightarrow \{I_1, I_2\}$
  - $q_2 \Rightarrow \{I_3, I_4\}$
  - $q_3 \Rightarrow \{I_5, I_6\}$
- $Q_2 \equiv \{q_1, q_2\}$ 
  - $q_1 \Rightarrow \{I_7, I_8\}$
  - $q_2 \Rightarrow \{I_4, I_9, I_{10}, I_{11}\}$
- $Q_3 \equiv \{q_1, q_2\}$ 
  - $q_1 \Rightarrow \{I_7, I_{12}\}$
  - $q_2 \Rightarrow \{I_{13}, I_{14}, \}$

Dado lo anterior planteado, por transitividad se observa que las CUEs  $Q_1, Q_2$  y  $Q_3$  requieren los siguientes elementos de datos:

- $Q_1 \equiv \{I_1, I_2, I_3, I_4, I_5, I_6\}$
- $Q_2 \equiv \{I_4, I_7, I_8, I_9, I_{10}, I_{11}\}$
- $Q_3 \equiv \{I_7, I_{12}, I_{13}, I_{14}\}$

EL valor de **relevancia de un elemento de dato** esta dado por la suma de **ecuación 4.7**, donde a partir de un elemento de dato se debe buscar el número de CUEs que lo solicitan, y cada una de esas CUEs aporta  $\frac{1}{k}$  a la relevancia de dicho elemento.

$$AR(Q_i) = \frac{1}{Anonimato(Q_i)} \quad (4.7)$$

Por lo tanto, obteniendo el aporte de relevancia con la **ecuación 4.7** se calcula el nivel de relevancia de un elemento de dato del siguiente modo:

$$R(I_i) = \sum_{Q \subset I_i} AR(Q_j) \quad (4.8)$$

Donde el nivel de relevancia para un elemento de dato  $R(I_i)$  es igual a la sumatoria del aporte de relevancia  $AR(Q_j)$  dado por la **ecuación 4.7** de las CUEs que contienen el elemento de dato  $Q_j \subset I_i$ .

Según los anteriores datos entregados se puede observar los siguientes valores de relevancia para los elementos de datos:

- $I_1 \in \{Q_1\} \implies R(I_1) = 0,3$
- $I_2 \in \{Q_1\} \implies R(I_2) = 0,3$
- $I_3 \in \{Q_1\} \implies R(I_3) = 0,3$
- $I_4 \in \{Q_1, Q_2\} \implies R(I_4) = 0,8$
- $I_5 \in \{Q_1\} \implies R(I_5) = 0,3$
- $I_6 \in \{Q_1\} \implies R(I_6) = 0,3$
- $I_7 \in \{Q_1, Q_3\} \implies R(I_7) = 1$
- $I_8 \in \{Q_2\} \implies R(I_8) = 0,5$
- $I_9 \in \{Q_2\} \implies R(I_9) = 0,5$
- $I_{10} \in \{Q_2\} \implies R(I_{10}) = 0,5$
- $I_{11} \in \{Q_2\} \implies R(I_{11}) = 0,5$
- $I_{12} \in \{Q_3\} \implies R(I_{12}) = 0,5$
- $I_{13} \in \{Q_3\} \implies R(I_{13}) = 0,5$
- $I_{14} \in \{Q_3\} \implies R(I_{14}) = 0,5$

Se considera que el elemento de dato más relevante es  $Q_7$ , de este modo debe tener mayor prioridad en el broadcasting. Por lo tanto, un broadcasting  $B$  se compone por los elementos de datos ordenados desde mayor a menor relevancia de la siguiente forma:

- $B = \{I_7, I_4, I_8, I_9, I_{10}, I_{11}, I_{12}, I_{13}, I_{14}, I_1, I_2, I_3, I_5, I_6\}$

Al igual que la técnica de envergadura y popularidad, en la técnica de relevancia no se repiten elementos de datos en el broadcasting. Entonces, los elementos de datos que tienen valores iguales de relevancia se ordenan según posición en la cola de consultas  $I_8, I_9, I_{10}, I_{11}, I_{12}, I_{13}, I_{14}$ .

A continuación se se presenta el algoritmo de scheduling de relevancia (R):

---

**Entrada:** respuestas requeridas por los usuarios  
**Salida:** Orden en que las respuestas a las CUEs serán transmitidas

- 1 R = arreglo de elementos de datos ponderados;
- 2 B = Conjunto de broadcasting;
- 3 **para**  $T = 1; T < \text{Repeticiones}; T++$  **hacer**
- 4 | Transmisión de broadcasting anterior;
- 5 **fin**
- 6 **para**  $i=1; i < \text{Numero de elementos de datos}; i++$  **hacer**
- 7 | Para cada Elemento de dato  $I_i$  calcular relevancia: **ecuación 4.8**;
- 8 |  $R[i] = P(I[i]);$
- 9 **fin**
- 10 **para**  $H=1; H < \text{Broadcast almacenados}; H++$  **hacer**
- 11 | **mientras**  $R \neq \emptyset$  **hacer**
- 12 | | E = Seleccionar desde R con probabilidad **ecuación 4.9** un elemento de dato;
- 13 | | b = agregar E a la cola;
- 14 | **fin**
- 15 | B = guardar broadcast b;
- 16 **fin**
- 17  $\text{Broadcast} =$  Seleccionar b de B según: certeza, stretch o Jitter;
- 18 **devolver**  $\text{Broadcast}$

---

Como conclusión, la técnica de relevancia presentada en el **algoritmo 5** tiene la misma forma que la técnica de envergadura y popularidad, con la diferencia de que con la técnica de relevancia se busca el nivel de requerimiento de los elementos de datos considerando que cada CUE aporta  $\frac{1}{KQ}$  al nivel de relevancia.

La técnica **R** es utilizada por un algoritmo de ordenamiento probabilista. De modo que el ponderador  $PR(I_i)$ , mostrado en la línea 3 del **algoritmo 5**, es considerada una probabilidad. Al igual que la técnica de envergadura y popularidad, la relevancia debe ser normalizada de modo que las probabilidades de cada uno de los elementos de dato sume 1. Es por ello que el valor  $R(I_i)$  de un elemento de dato es normalizada de la siguiente forma:

$$PR(I_j) = \frac{R(I_j)}{\sum_{j=1}^N R(I_j)} \quad (4.9)$$

Donde  $R(I_i)$  es dado por la **ecuación 4.8**,  $N$  es el número de CUEs que contienen al elemento de dato  $I_j$  y  $\sum_{j=1}^N R(I_j)$  es la suma de todas las probabilidad de los elementos

de datos requeridos por las CUEs de una cola de consultas. Por ejemplo, dadas las CUEs  $Q' = \{Q_1, Q_2, Q_3\}$  se obtiene que :

- $PP(I_1) = \frac{0,3}{6,8} = 0,048$
- $PP(I_2) = \frac{0,3}{6,8} = 0,048$
- $PP(I_3) = \frac{0,3}{6,8} = 0,048$
- $PP(I_4) = \frac{0,8}{6,8} = 0,12$
- $PP(I_5) = \frac{0,3}{6,8} = 0,048$
- $PP(I_6) = \frac{0,3}{6,8} = 0,048$
- $PP(I_7) = \frac{1}{6,8} = 0,17$
- $PP(I_8) = \frac{0,5}{6,8} = 0,07$
- $PP(I_9) = \frac{0,5}{6,8} = 0,07$
- $PP(I_{10}) = \frac{0,5}{6,8} = 0,07$
- $PP(I_{11}) = \frac{0,5}{6,8} = 0,07$
- $PP(I_{12}) = \frac{0,5}{6,8} = 0,07$
- $PP(I_{13}) = \frac{0,5}{6,8} = 0,07$
- $PP(I_{14}) = \frac{0,5}{6,8} = 0,07$

Según los anteriores datos de los elementos, un broadcasting ordenado con la técnica **R** indica que los elementos de datos  $\{I_8, I_9, I_{10}, I_{11}, I_{12}, I_{13}, I_{14}\}$  tienen un 7% de posibilidades de estar en las primeras posiciones, los elementos de datos  $\{I_1, I_2, I_3, I_5, I_6\}$  tienen un 4,8% de posibilidades de estar en las primeras posiciones, el elemento de dato  $I_4$  tiene 12% de posibilidades de estar en las primeras posiciones del broadcasting y el elemento de dato  $I_7$  tiene una probabilidad del 17% de estar en las primeras posiciones del broadcasting.

Por otro lado, se presenta la técnica **RxW** la cual agrega al valor de relevancia el tiempo de espera del elemento de dato. Para ello se selecciona el mayor tiempo de espera de las CUEs que requieren un elemento de dato. De este modo se obtienen los siguientes datos para las CUES  $Q_1, Q_2, Q_3$ :

- $W(Q_1) = 8$
- $W(Q_2) = 10$
- $W(Q_3) = 12$

Entonces, dados los anteriores datos de tiempo de espera, los elementos de datos contiene los siguientes tiempo de espera:

- $W(I_1) = 8$
- $W(I_2) = 8$
- $W(I_3) = 8$
- $W(I_4) = 10$
- $W(I_5) = 8$
- $W(I_6) = 8$
- $W(I_7) = 12$
- $W(I_8) = 10$
- $W(I_9) = 10$
- $W(I_{10}) = 10$
- $W(I_{11}) = 10$
- $W(I_{12}) = 12$
- $W(I_{13}) = 12$
- $W(I_{14}) = 12$

El múltiplo de las ponderaciones  $PR(I_i) \cdot W(I_i)$  como probabilidad para un algoritmo de ordenamiento probabilista es distinto de 1 al combinarlo con varios elementos de datos. Es por ello que para calcular **PRxW** se requiere normalizar el múltiplo entre la ponderación probabilista de la popularidad y el tiempo de espera del elemento de dato. De este modo se obtiene la siguiente probabilidad:

$$PRxW(I_j) = \frac{PR(I_j) \cdot W(I_j)}{\sum_{j=1}^N PR(I_j) \cdot W(I_j)} \quad (4.10)$$

Donde  $PR(I_i)$  esta dada por la **ecuación 4.9**,  $N$  es el número de CUEs que contienen al elemento de dato  $I_j$  y  $\sum_{j=1}^N PR(I_j) \cdot W(I_j)$  es la suma de todas las probabilidades  $PR(Q_i)$  de las CUEs de una cola de espera con su respectivo tiempo de espera  $W(I_j)$ . Por ejemplo, dados los anteriores elementos de datos  $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11}, I_{12}, I_{13}, I_{14}$ , según la **ecuación 4.10** se obtiene que:

- $PRxW(I_1) = \frac{0,048 \cdot 8}{10,52} = 0,036$
- $PRxW(I_2) = \frac{0,048 \cdot 8}{10,52} = 0,036$
- $PRxW(I_3) = \frac{0,048 \cdot 8}{10,52} = 0,036$
- $PRxW(I_4) = \frac{0,12 \cdot 10}{10,52} = 0,11$
- $PRxW(I_5) = \frac{0,048 \cdot 8}{10,52} = 0,036$
- $PRxW(I_6) = \frac{0,048 \cdot 8}{10,52} = 0,036$
- $PRxW(I_7) = \frac{0,17 \cdot 12}{10,52} = 0,209$
- $PRxW(I_8) = \frac{0,07 \cdot 10}{10,52} = 0,066$
- $PRxW(I_9) = \frac{0,07 \cdot 10}{10,52} = 0,066$
- $PRxW(I_{10}) = \frac{0,07 \cdot 10}{10,52} = 0,066$
- $PRxW(I_{11}) = \frac{0,07 \cdot 10}{10,52} = 0,066$
- $PRxW(I_{12}) = \frac{0,07 \cdot 12}{10,52} = 0,079$
- $PRxW(I_{13}) = \frac{0,07 \cdot 12}{10,52} = 0,079$
- $PRxW(I_{14}) = \frac{0,07 \cdot 12}{10,52} = 0,079$

Según los anteriores datos de los elementos, un broadcasting ordenado con la técnica **R-W** indica que los elementos de datos  $\{I_1, I_2, I_3, I_5, I_6\}$  tienen un 3,6% de posibilidades de estar en las primeras posiciones, los elementos de datos  $\{I_8, I_9, I_{10}, I_{11}\}$  tienen un 6,6% de posibilidades de estar en las primeras posiciones, los elementos de datos  $\{I_{12}, I_{13}, I_{14}\}$  tienen un 7,9% de posibilidades de estar en las primeras posiciones, el elemento de dato  $I_4$  tiene

11 % de posibilidades de estar en las primeras posiciones del broadcasting y el elemento de dato  $I_7$  tiene 20,9 % de posibilidades de estar en las primeras posiciones del broadcasting.

A continuación se se presenta el algoritmo de scheduling de relevancia con tiempo de espera(PxW):

---

**Algoritmo 5:** ALGORITMO DE RELEVANCIA CON TIEMPO DE ESPERA(RxW)

---

**Entrada:** respuestas requeridas por los usuarios

**Salida:** Orden en que las respuestas a las CUEs serán transmitidas

```

1 R = arreglo de elementos de datos ponderados;
2 B = Conjunto de broadcasting;
3 para  $T = 1; T < Repeticiones; T++$  hacer
4   | Transmisión de broadcasting anterior;
5 fin
6 para  $i=1; i < Numero\ de\ elementos\ de\ datos; i++$  hacer
7   | Para cada Elemento de dato  $I_i$  calcular relevancia: ecuación 4.9;
8   |  $R[i] = P(I[i]);$ 
9 fin
10 para  $H=1; H < Broadcast\ almacenados; H++$  hacer
11   | mientras  $R \neq \emptyset$  hacer
12     | E = Seleccionar desde R con probabilidad ecuación ?? un elemento de dato;
13     | b = agregar E a la cola;
14   | fin
15   | B = guardar broadcast b;
16 fin
17 Broadcast = Seleccionar b de B según: certeza, stretch o Jitter;
18 devolver Broadcast
```

---

El **algoritmo 6** difiere del **algoritmo 5** en la línea 12. Donde se utiliza como ponderador el producto normalizado de la probabilidad de relevancia  $PR(I_i)$  y el tiempo de espera  $W(I_i)$ .

## 4.5. Algoritmos de ajuste para las estrategias de scheduling

A continuación se presentan en mayor detalle la fase de ajuste de los cronogramas creados con las técnicas **ExW**, **PxW** y **RxW**. Se debe considerar que los broadcasting se generan en base a un algoritmo de selección probabilista, entonces, pensando en que los anteriores algoritmos de scheduling se pueden repetir y entregar varios broadcasting distintos, es a estos a quienes se les evalúa ya sea su nivel de certeza, stretch o jitter.

El **algoritmo de ajuste basado en la certeza (C)** tiene como objetivo seleccionar el broadcast, que dentro de los T primeros elementos de datos, cuente con la mayor cantidad de elementos de datos que pertenecen a CUEs con bajo anonimato y que requieren pocos elementos.

La certeza calcula el valor de los T primeros buckets de un broadcast construido con un algoritmo de scheduling. A continuación se presenta el algoritmo de ponderación de la certeza.

Para calcular la ponderación de la certeza se calcula el número de consultas de rango que requieren un determinado elemento de dato y posteriormente, de esas consultas de rango seleccionadas, se calcula la sumatoria del aporte basado en el k-anonimato de las CUE a la que pertenece. A continuación se muestra la fórmula:

$$\blacksquare \text{ certeza}(P) = \left( \sum_{i=1}^P \left( \sum_{j=1}^Q \frac{1}{K_{Q_j}} \left( \sum_{k=1}^{k_{Q_j}} \frac{\text{Aporte del Item } q_k}{|q_k|} \right) \right) \right)$$

Donde el *Aporte del Item*  $q_k$  corresponde a la siguiente función múltiple:

$$\blacksquare \text{ Aporte del Item } q_k = \begin{cases} 1, & \text{si } I_j \in q_{Q_j} \\ 0 & \text{de otra forma} \end{cases}$$

El **algoritmo de selección basado en stretch (S)** tiene como objetivo seleccionar el broadcast, que dentro de los T primeros elementos de datos que responden una consulta de rango son difundidos en un lapso de tiempo proporcional al tamaño de la consulta de rango. En otras palabras, esto significa que las CUEs serán respondidas en función a su tamaño. Por lo tanto, se estima que las consultas de rango que solicitan un mayor número de elementos de datos deben estar dispuestas a esperar una mayor cantidad de tiempo para recibir todas las respuestas solicitadas; contando las superfluas.

El stretch calcula el valor para los T primeros buckets de un broadcast construido con un algoritmo de scheduling probabilista. A continuación se presenta el algoritmo de ponderación de stretch.

Para calcular la ponderación de stretch se busca la consultas de rango que mayor stretch tienen, de modo que, si el mayor stretch es bajo significa que los elementos de datos que responden consultas de rango lo hacen tan pronto como sea posible. A continuación se muestra la fórmula:

$$\blacksquare \text{ Stretch}(q_{\text{completada}}) = \frac{\text{tiempo de respuesta}(q_{\text{completada}})}{\text{tiempo servicio}(q_{\text{completada}})}$$

Donde el tiempo de servicio es el tiempo para completar una consulta como si ella fuera la única en ser respondida. Análogamente, es el largo de la consulta. Mientras que el tiempo

de respuesta es desde que comienza el broadcast hasta que se recibe el último elemento de dato.

El **algoritmos de selección basado en jitter (J)** tiene como objetivo seleccionar el broadcast, que dentro de los T primeros elementos de datos que responden una subconsulta de rango son difundidos en transcurso de tiempos similares. En otras palabras, esto significa que las consultas recibirán los elementos de datos con latencias tan similares como sea posible. Por lo tanto, se estima que las consultas de rango reciben los elementos de datos equánimemente respecto al tiempo, es decir, cuando una subconsulta de rango recibe una respuesta y después de tres recibe la segunda, se espera que en las próximas tres se reciba la tercera. En ese sentido, importa que el orden del sub-broadcast sea equánime respecto a la distancia de los elementos de datos que responden una subconsulta de rango.

El jitter calcula el valor de jitter de los T primeros buckets de un broadcasting construido con un algoritmo de scheduling probabilista. A continuación se presenta el algoritmo de ponderación de jitter.

Para calcular la ponderación de jitter se busca la subconsulta de rango que mayor jitter tienen, de modo que, si el mayor jitter es bajo significa que los elementos de datos que responden consultas de rango lo hacen en lapsos de tiempos tan similares como sea posible. A continuación se muestra la fórmula:

$$\blacksquare \text{ Jitter}(q_{\text{completada}}) = \frac{\text{Suma de distancias}(q_{\text{completada}})}{\text{numero de elementos}(q_{\text{completada}})-1}$$

A modo de conclusión, el hecho de medir la certeza, stretch y jitter de una sola parte del broadcast se debe a que los primeros T elementos de datos difundidos son los que se mantendrán en el tiempo, siempre y cuando, los usuarios se concentren en un determinado sector del plano y los traslados hacia otros sectores se realicen de forma gradual.

# Capítulo 5

## Evaluación del sistema

En ésta sección se realiza una evaluación de las principales características de algoritmos de scheduling tales como:

- Algoritmo de envergadura con tiempo de espera.
- Algoritmo de popularidad con tiempo de espera.
- Algoritmo de relevancia con tiempo de espera.

En combinación con los anteriores algoritmos de scheduling se aplican las siguientes estrategias de ajuste:

- Ajuste por certeza.
- Ajuste por jitter.
- Ajuste por stretch.

El ambiente en que se realiza la simulación del sistema se divide en varias parte, entre ellas las mas importantes son:

- **Usuario:** este lleva a cabo acciones tales como moverse en el plano, observar el broadcast, construir y enviar consultas al servidor LBS, entre otras. El movimiento del usuario a través del plano se realiza en todas las direcciones cardinales; norte, sur, este, oeste, noreste, noroeste, sureste y suroeste. Cada una de esas direcciones tienen distintas distribuciones. Por ejemplo, en el caso cuando los usuarios se mueven de forma aleatorias través de todo el plano, la distribución es uniforme para cada uno de los puntos

cardinales, por otro lado, cuando el usuario se encuentra con el borde del plano, se aplica un cambio de las probabilidades de movimiento de modo que se privilegia el movimiento hacia el lado opuesto del borde. Cuando el usuario observa el broadcast debe almacenar los elementos de datos que responden sus consulta y el primer elemento que recibe es el que se mantiene como elemento recogido del broadcast.

- **Servidor:** éste lleva a cabo operaciones tales como construir un broadcast en función de las consultas que envían los usuarios. Para llevar a cabo estas acciones, el servidor debe realizar una búsqueda de los elementos de datos que están relacionados con las consultas que debe procesar. Una vez terminado esto, el servidor debe construir diferentes estructuras de datos para registrar todos los datos de la simulación. Para la construcción del broadcast el servidor debe utilizar alguna de las técnicas de schedule en combinación con un algoritmo de selección siempre y cuando sea necesario. Luego de esto el servidor debe truncar el broadcast para comenzar la difusión de un sub-broadcast del tamaño especificado con anterioridad.

En esta investigación se realiza la simulación de un ambiente en tiempo real, en el sentido que se simula el movimiento de los usuarios, su comportamiento a la hora de generar y enviar consultas a un servidor LBS, el modo en que el usuario observa los broadcast, el modo en que el servidor construye y difunde los broadcast. En otras palabras, en vez de utilizar distribuciones para generar la posición de los usuarios y las consultas en un determinado momento, se establece un comportamiento más dinámico de los usuarios y a partir de ese comportamiento se van generando consultas.

Sin embargo, también se utilizan distribuciones para algunas operaciones del sistema. Por ejemplo, en el momento en que el usuario se debe mover se utiliza una distribución uniforme. Para establecer la posición inicial de los usuarios se utiliza la distribución zipf ( $S=2$ ). Finalmente, para decidir quiénes son los usuarios que envían consultas al servidor LBS se utiliza una distribución uniforme.

Con respecto al tiempo utilizado para medir el rendimiento de las diferentes técnicas de schedule, se utiliza un bucket. Entonces, una unidad de tiempo equivale a un bucket y un bucket equivale a la difusión de un elemento de dato, así como a un paso de un usuario. Por ejemplo, una simulación que demora en terminar un tiempo de 7500 significa que se difundieron 7500 elementos de datos y los usuarios se movieron la misma cantidad de pasos.

A continuación se presenta el ambiente en el que fueron realizadas las pruebas, mostrando los parámetros que son predefinidos y los que no son predefinidos, además se muestra de forma general el procedimiento para realizar la simulación y finalmente se presenta el

lenguaje de programación y las librerías mas importantes utilizadas. Luego se muestran los resultados tales como el tiempo de respuesta, la utilidad y efectividad, la justicia, la obsolescencia, privacidad y carga de trabajo de los que en esta investigación se consideran las técnicas de schedule mas representativas.

## 5.1. Ambiente de pruebas

En esta simulación se simula un sistema que construye un broadcast con algoritmos de schedule. Para ello se crea un sistema que genera el movimiento de los usuarios y mientras ellos se mueven en un plano predefinido con un número de celdas (filas y columnas) y un tamaño (largo y ancho de una celda), van creando consultas de rango. El usuario antes de enviar su CUE debe observar el broadcast para ver si las respuestas están siendo difundidas. Si el usuario las encuentra, antes de esperar a que se comience a transmitir un nuevo broadcast, crea una nueva CUE. Por otro lado, cuando no encuentra los elementos debe esperar a que se termine de difundir el broadcast para poder enviarla.

El simulador esta construido de modo tal que se restringen parámetros como:

- El número de consultas creada por usuarios.
- El anonimato de las consultas.
- El tamaño de los sub-broadcast.
- El número de veces que se debe repetir la difusión de un broadcast.
- El número de usuarios del sistema.
- El número de puntos de interés del sistema.
- El porcentaje de usuarios que pueden enviar consultas al servidor LBS.
- El movimiento de los usuarios.

Sin embargo, también existen parámetros que no son controlados por el simulador, como por ejemplo:

- Las celdas seleccionadas para anonimizar una consulta.
- El número de CUEs (consultas anonimizadas) que los usuarios envían al servidor LBS.

- El tamaño de los broadcast.

La selección de celdas para anonimizar una consulta esta relacionada con trabajos que investigan la forma de construir una consulta de ubicación encubierta. En ese sentido, esta investigación queda abierta a la posibilidad de utilizar alguna de las técnicas presentadas en la literatura (Sweeney, 2002).

Por otro lado, cumple un rol importante en esta investigación el hecho de que no se pueda controlar cuáles son las CUEs que los usuario envían al servidor LBS, ya que de este modo se puede saber el nivel de privacidad de los usuarios, el tiempo de respuesta, entre otros parámetros.

Es un hecho de que no se pueda restringir el tamaño de los broadcast ya que que de otro modo podría haber la posibilidad de causar el problema de la hambruna, es decir, si se envía solo una parte del broadcast existe la posibilidad de que usuarios no puedan completar su CUE, entonces, todos los elementos solicitados deben ser enviados solo por el hecho de que el servidor LBS no puede pronosticar cuáles son los elementos que serán utilizados por los usuarios. Por otro lado, las CUEs que se envían al servidor LBS tienen diferentes tamaños (número de elementos de datos requeridos) ya que esto depende de la densidad de puntos de interés del sector donde se crean y las celdas utilizadas para anonimizar las consultas.

Una vez construido un broadcast, éste se difunde de modo que todos los usuarios del sistema pueden observar los elementos de datos que se están transmitiendo.

Éste simulador fue programado con el lenguaje de programación Python con su versión 2.7.17, algunas de las librerías que se utilizaron son:

- numpy = 1.16.4
- pandas = 0.24.2
- plotly = 4.0.0

Las cuales fueron utilizadas para realizar el trabajo relacionado con las distribuciones, promedios, entre otras operaciones matemáticas.

## 5.2. Tiempo de respuesta

El tiempo de respuesta se define como el lapso transcurrido entre el momento en que una consulta es creada hasta el momento en que es respondida. En este trabajo se mide el tiempo de respuesta de diferentes algoritmos de scheduling combinados con algoritmos de selección. Como se menciona anteriormente una unidad de tiempo equivale a un bucket, como un bucket equivale a un elemento de dato difundido y un paso de un usuario.

El tiempo de respuesta se calcula como el promedio de los tiempos de respuesta de las consultas de los usuarios. Donde a mayor tiempo de respuesta promedio se observa que los usuarios tuvieron que esperar una mayor tiempo para responder sus consultas.

En este trabajo, para analizar de forma más precisa el tiempo de respuesta, se divide en tres diferentes tipos de tiempos de repuesta:

- Tiempo de respuesta del servidor.
- Tiempo de respuesta del broadcast.
- Tiempo de respuesta total.

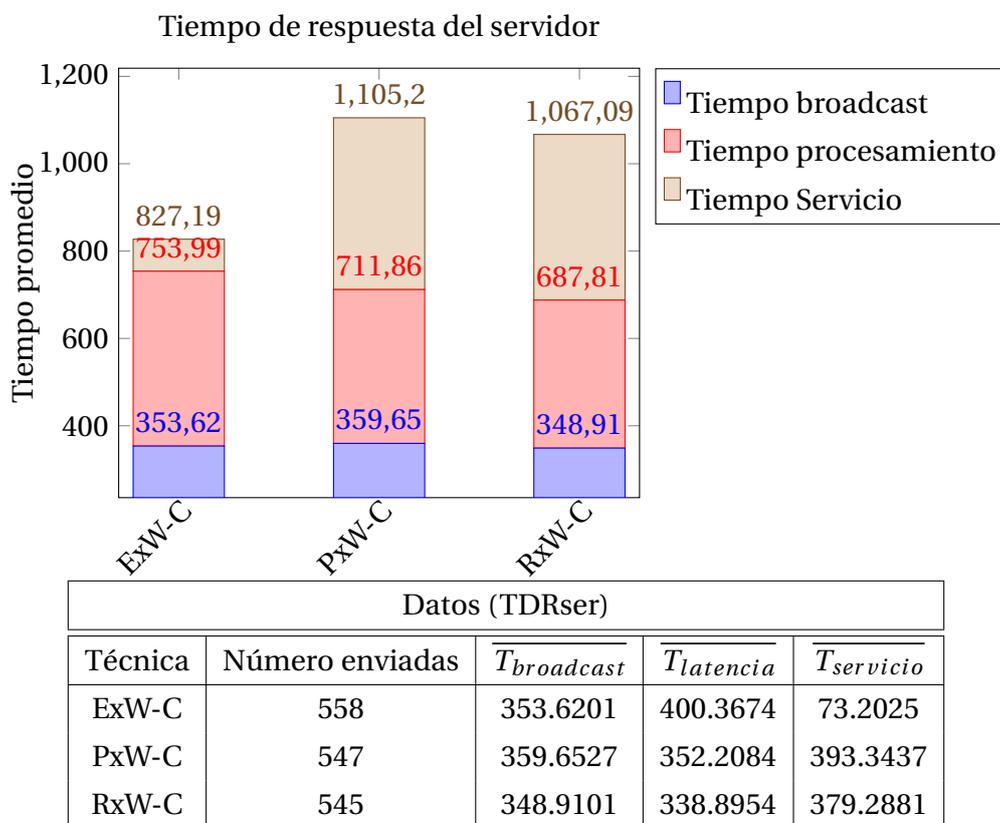
El tiempo de respuesta del servidor se mide sólo para las CUEs que los usuarios envían al servidor LBS. Sin embargo, a pesar de que para todos los experimentos el movimiento de los usuarios y el tamaño de las CUEs son iguales, al variar los algoritmos de scheduling, llegan deferentes cantidades de consultas al servidor LBS. Por lo tanto, para tener una visión más precisa del tiempo de respuesta, este parámetro se forma como la sumatoria del tiempo de broadcast , el tiempo de procesamiento y el tiempo de servicio de las CUEs enviadas al servidor LBS.

El tiempo de respuesta del broadcast se mide para las CUEs que no son enviadas al servidor LBS. Entonces, como se menciona anteriormente, dada las condiciones, en todas las simulaciones hay distintos números de consultas que no se envían al servidor. Sin embargo, el tiempo de respuesta es sólo el tiempo en que el usuario se tarda en encontrar la respuesta en el broadcast.

Finalmente, el tiempo de respuesta total se mide tanto para las consultas que son enviadas como las que no son enviadas al servidor LBS. Por lo tanto, es aquí en donde se puede obtener una cifra más real del tiempo de respuesta.

### 5.2.1. Algoritmo probabilista con tiempo de espera cambiando los ritmos de schedule

A continuación se presentan los Cuadro 5.1 , 5.2 y 5.3 que muestran el tiempo de respuesta de las CUEs que fueron enviadas al servidor LBS para los algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera y los algoritmos de selección de certeza, donde se mantiene el k-anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.



**Cuadro 5.1:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

El anterior Cuadro 5.1 de tiempo de respuesta del servidor muestra los algoritmos de envergadura, de popularidad y relevancia probabilista con tiempo de espera trabajando bajo el algoritmo de selección de certeza.

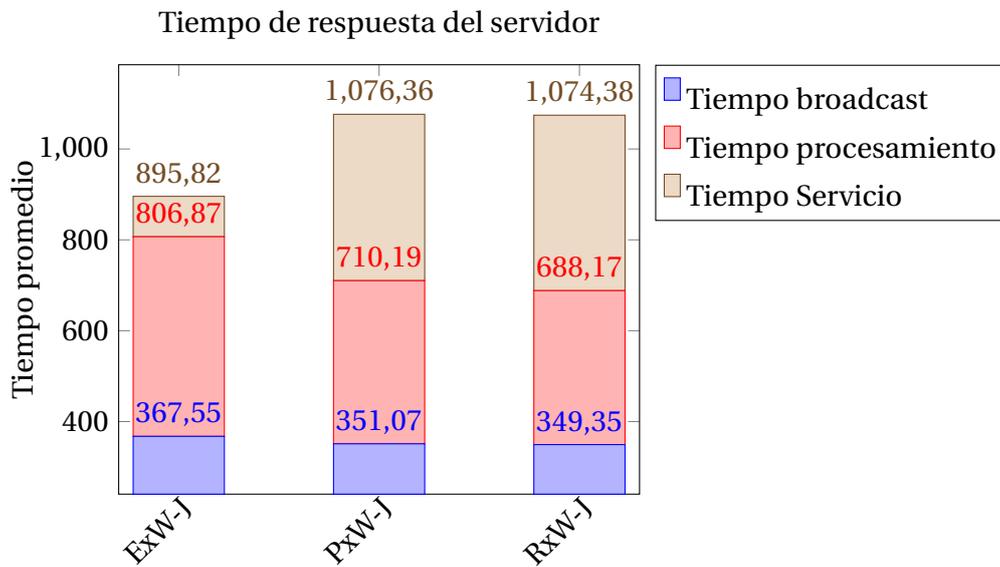
Como se puede observar, el tiempo de broadcast de las tres técnicas se muestra similar entre 348 y 360. Esto significa que las independiente de las técnicas de planificación, al aplicar

la selección por certeza no se afecta el tiempo de búsqueda de las respuestas en un broadcast, ésto se puede entender por la similaridad que tienen respecto a sus tamaños ya que los usuarios están pidiendo siempre los mismos datos.

Por otro lado, el tiempo de procesamiento presenta cambios significativos al variar el algoritmo de scheduling. Por ejemplo, con la técnica de relevancia, la cual presenta el menor tiempo de procesamiento, se puede observar que mejora en 66.18 puntos menos que la técnica de envergadura; la cual es la de peor tiempo de procesamiento.

Finalmente, la vista macro del **Cuadro 5.1** muestra que el tiempo de servicio de las técnicas de scheduling de popularidad y relevancia empeoran con creces en el rendimiento del tiempo de servicio y por lo tanto el tiempo de espera empeora en el peor caso en 278.01 puntos. Este comportamiento se debe principalmente porque tanto la técnica de popularidad como la de relevancia tienen una organización del broadcast de forma más globalizada, es decir, el broadcast se planifica por elemento de datos, dando la oportunidad de que los elementos de datos sean igual de importante entre ellos.

Observando los **Cuadro 5.2** y **Cuadro 5.3** mostrados a continuación se observa el mismo comportamiento esto significa que sin importar el algoritmo de selección aplicado, el hecho de que los elementos de datos sean individualizados el algoritmos de selección no tienen un mayor impacto en mejorar el tiempo de servicio de las consultas que son enviadas al servidor LBS.



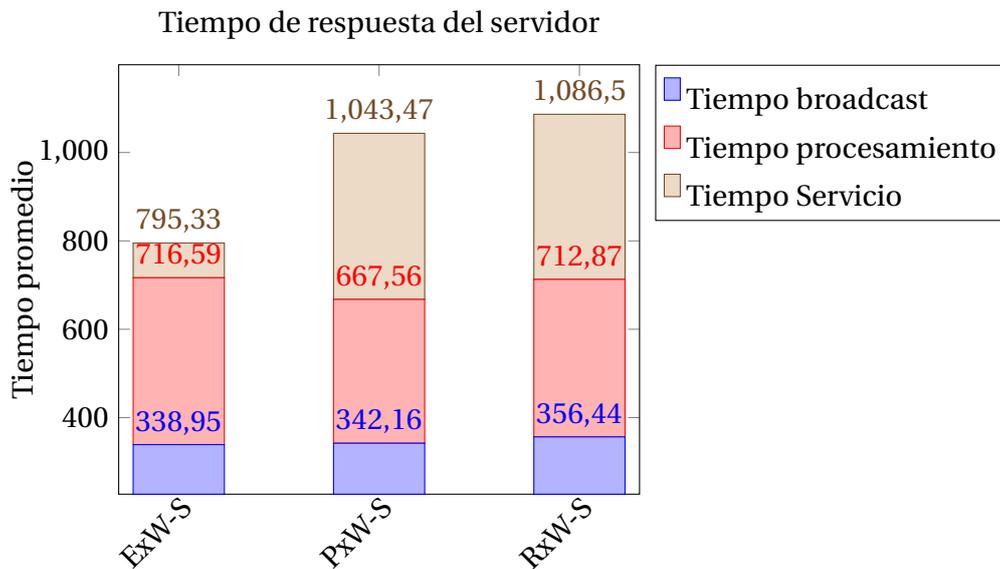
Datos (TDRser)				
Técnica	Número enviadas	$\overline{T}_{broadcast}$	$\overline{T}_{latencia}$	$\overline{T}_{servicio}$
ExW-J	550	367.5509	439.3218	88.9509
PxW-J	544	351.0735	359.1140	366.1765
RxW-J	526	349.3460	338.8232	386.2072

**Cuadro 5.2:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]



Datos (TDRser)				
Técnica	Número enviadas	$\overline{T}_{broadcast}$	$\overline{T}_{latencia}$	$\overline{T}_{servicio}$
ExW-S	532	338.9455	377.6429	78.7368
PxW-S	570	342.1596	325.4053	375.9018
RxW-S	549	356.4372	356.4281	373.6375

**Cuadro 5.3:**

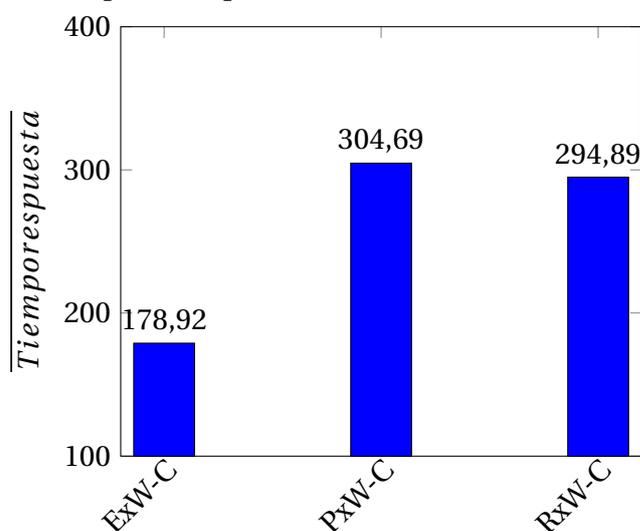
ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

A continuación se presentan los **Cuadro 5.4 ,5.5 , 5.6** que muestran el **tiempo de respuesta de las consultas que no son enviadas al servidor LBS** para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.

Tiempo de respuesta CUEs resueltas en el broadcast



Datos (TDRbro)			
Técnica	CUEs creadas	CUEs enviadas	CUEs no enviadas
ExW-C	1400	558	842
PxW-C	1400	547	853
RxW-C	1400	545	855

**Cuadro 5.4:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

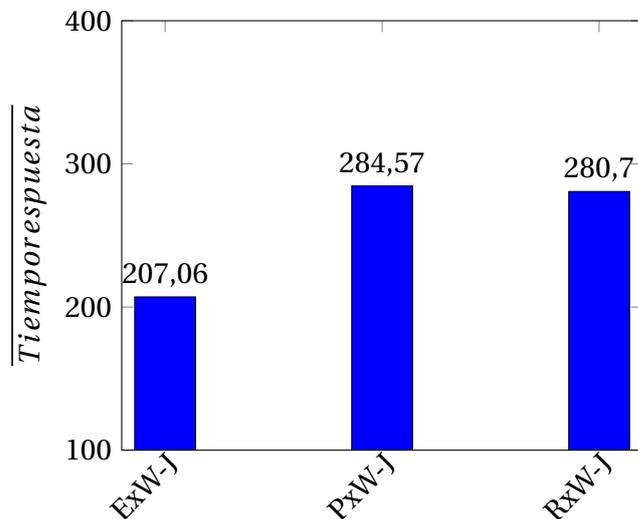
El anterior **Cuadro 5.4** de tiempo de respuesta del servidor muestra los algoritmos de envergadura, de popularidad y relevancia probabilista con tiempo de espera trabajando bajo el algoritmo de selección de certeza.

Como se puede observar, el tiempo de respuesta promedio es desde el momento en que los usuarios crean su consulta hasta que encuentran los elementos de datos para responderla. Las tres técnicas se muestran con distinto valores, sin embargo la técnica de envergadura presenta notables diferencias, teniendo 125.77 puntos de tiempo promedio menos que el peor de los casos (APPT-ASC). Éstas diferencias se debe a la misma situación que se presenta con los anteriores **Cuadro 5.1** , **5.2** y **5.3**; el algoritmo de envergadura ordena el broadcast agrupando los elementos de datos solicitados por las consultas mientras que la popularidad y relevancia los ordenan agrupando los elementos de datos de forma individual.

En los siguientes **Cuadro 5.5** y **Cuadro 5.6** se observa que el algoritmo de selección por jitter aumenta el tiempo de respuesta de la técnica de scheduling de envergadura. Sin

embargo, las técnicas de popularidad y relevancia presentan menores tiempo de respuesta al aplicar la técnica de selección por jitter. Esto se explica por el hecho de que la técnica de selección por jitter igualan, tanto como sea posible, la distancia de los elementos de datos, y por lo tanto, en el caso de la técnica de envergadura que ordena los broadcast agrupando los elementos de las CUEs los afecta de forma negativa en el tiempo de respuesta, ya que aumenta la distancia entre los elementos de datos. Mientras que bajo el mismo contexto, cuando se aplica la selección por jitter para las técnicas de popularidad y relevancia, igualar la distancia afecta de forma positiva el tiempo de respuesta, ya que los elementos que responden a una consulta y que están más distanciados dentro del broadcast, terminan estrechando las diferencias de sus distancias.

Tiempo de respuesta CUEs resueltas en el broadcast



Datos (TDRbro)			
Técnica	CUEs creadas	CUEs enviadas	CUEs no enviadas
ExW-J	1400	550	850
PxW-J	1400	544	856
RxW-J	1400	526	874

**Cuadro 5.5:**

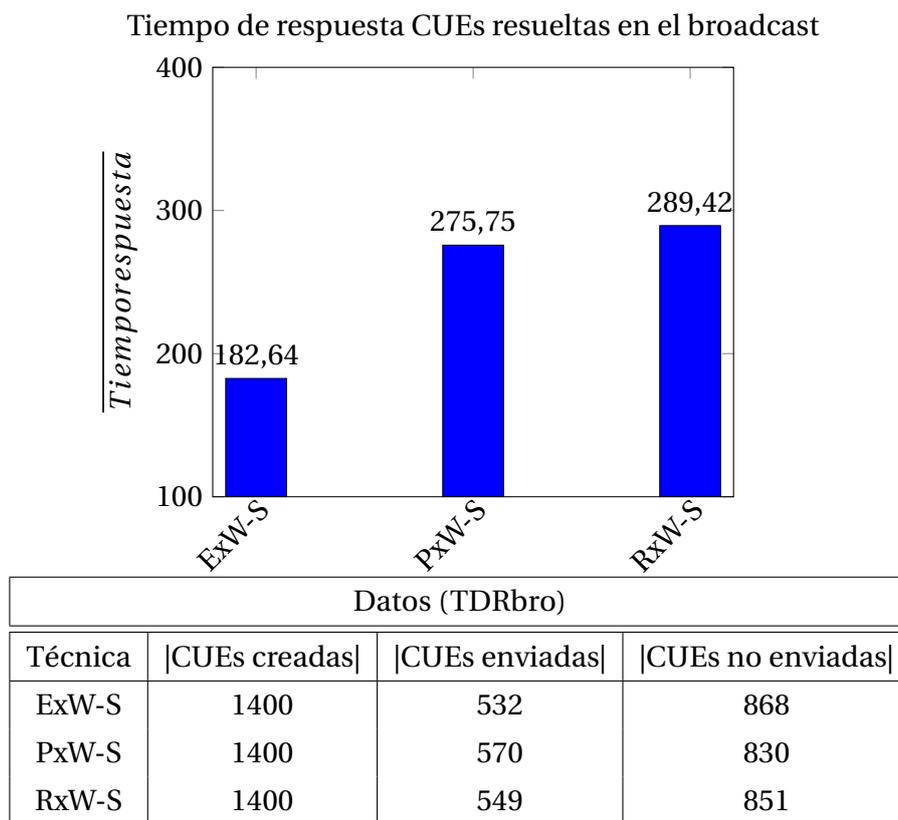
ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Finalmente, con la técnicas de selección del **Cuadro 5.6** de selección por stretch se observa que el algoritmo de scheduling por relevancia es aquel que presenta los peores tiempos de espera. Principalmente, ajustar por stretch significa que los usuarios esperan la

respuesta según el largo de su consulta, es decir, si los usuarios envían una CUE más larga al servidor LBS deben esperar más tiempo que los que envían una CUE más corta. Sin embargo, este hecho no justifica el aumento del tiempo de respuesta de la técnica de scheduling por relevancia. No obstante, el hecho de que se aplique una solución probabilista para ordenar los broadcast justifica el cambio, que por lo demás, no es tan radical, ya que aumenta sólo en 9 puntos.



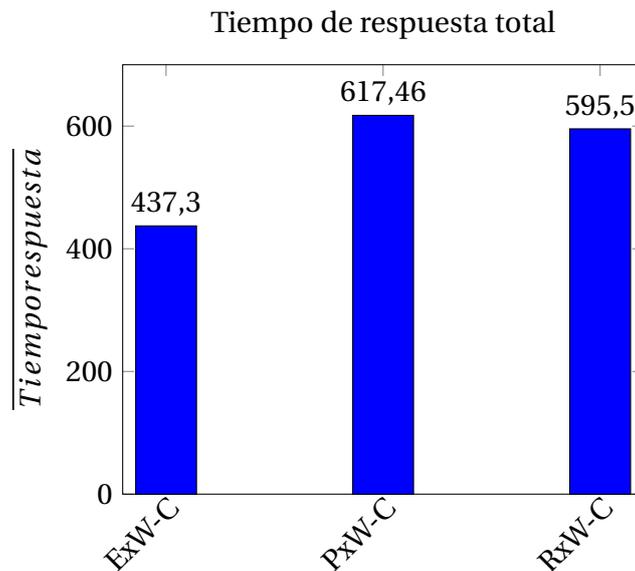
**Cuadro 5.6:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

A continuación se presenta los **Cuadro 5.7, 5.8 y 5.9** que muestran el **tiempo de respuesta de todas las consultas** del sistema para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.

**Cuadro 5.7:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

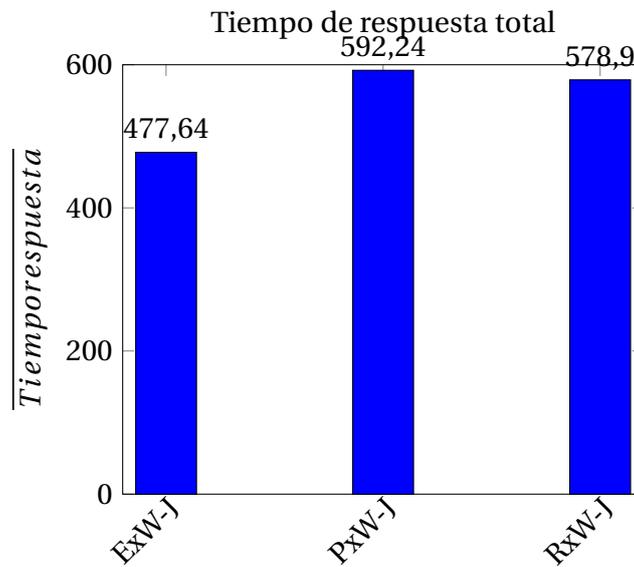
PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

El anterior **Cuadro 5.7** de tiempo de respuesta total muestra los algoritmos de envergadura, popularidad y relevancia probabilista con tiempo de espera trabajando bajo el algoritmo de selección de certeza.

El tiempo de respuesta con la técnica de scheduling de envergadura presenta el menor tiempo de respuesta total. Esto se debe a que tanto en el tiempo de respuesta de las CUEs que son enviadas como de las consultas que no son enviadas siempre se obtuvieron mejores resultados.

Finalmente, en los **Cuadro 5.8** y **Cuadro 5.9** a continuación se presenta el mismo comportamiento respecto al tiempo de respuesta.

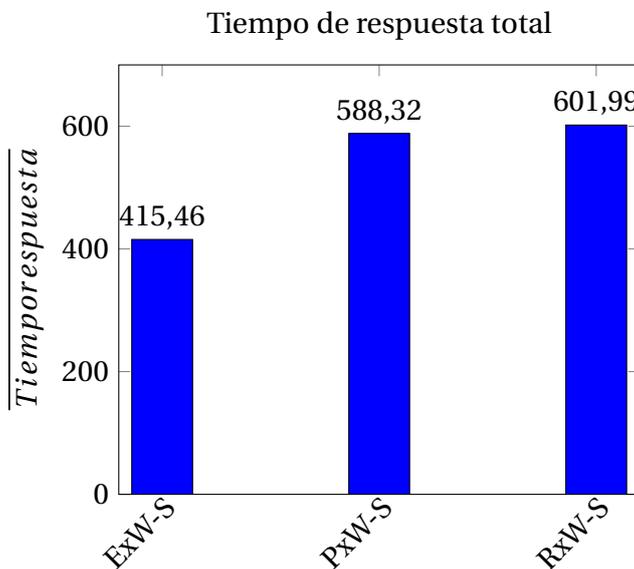


**Cuadro 5.8:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]



**Cuadro 5.9:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

A modo de conclusión, los mejores tiempo de respuesta son aquellos que presentan un menor promedio. Sin embargo, no deja de ser importante destacar que las diferentes técnicas

de scheduling y no son peores o mejores sólo con los resultados del tiempo de respuesta. Ésto significa que este tipo de técnicas, que optimizan el tiempo de respuesta, se pueden aplicar en los casos en que los usuarios requieren responder sus consultas lo más rápido posible o cuando los usuarios, en su gran mayoría, están optando por reducir su anonimato.

### 5.3. Utilidad y efectividad

La utilidad es una medida de rendimiento del broadcast respecto a los algoritmos de scheduling. En otras palabras, se mide el número de elementos de datos utilizados por las consultas respondidas por cada uno de los broadcast difundidos por el servidor LBS sobre el largo de los mismos broadcast difundidos.

Para representar la utilidad de los broadcast se evalúan las consultas que son creadas en el momento en que se comienzan a difundir los elementos de datos. Entonces una vez que se difunde el broadcast se evalúa, cuantos elementos de datos fueron utilizados por las consultas que fueron creadas antes de la difusión. Obteniendo ese valor, se calcula el largo del broadcast difundido y se obtiene la razón de utilidad.

El hecho de utilizar los elementos creados antes de difundir el broadcast es más preciso que medir el número de elementos utilizados por las consultas que son respondidas en el broadcast que se está difundiendo. Esto se debe principalmente a que cuando se crean consultas el usuario debe primero observar lo que se está difundiendo antes de enviar una CUE al servidor LBS. Entonces, si se midiera la utilidad de las consultas que son respondidas en un broadcast, se medirían únicamente la utilidad para las CUEs que son enviadas al servidor LBS. En teoría esta medida es válida, sin embargo, entre las distintas simulaciones es poco probable que lleguen la misma cantidad de CUEs al servidor LBS; por otro lado, siempre se crean la misma cantidad de consultas.

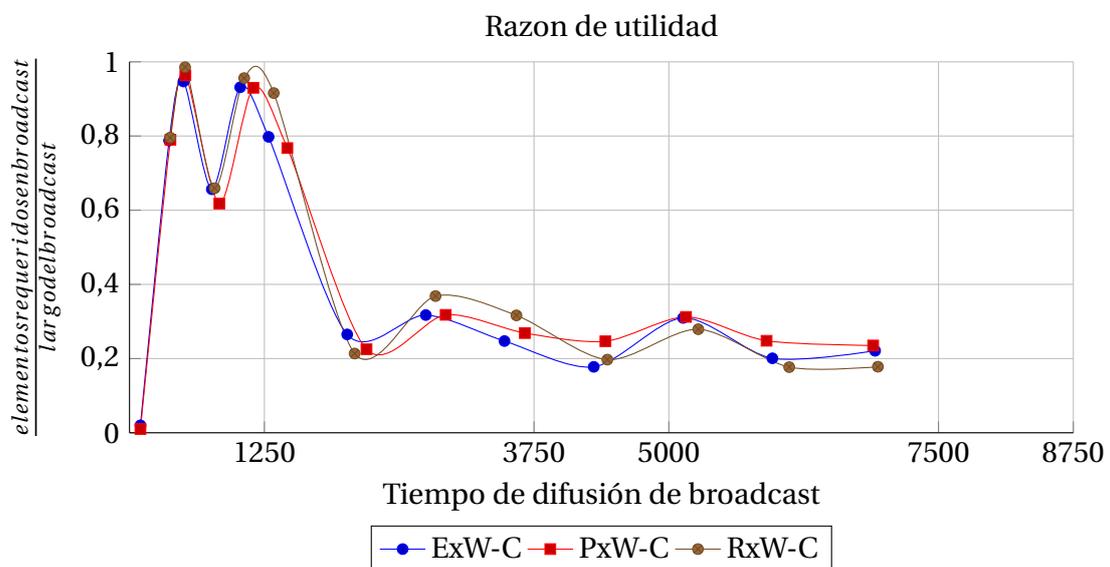
La efectividad es otra de las medidas de rendimiento del broadcast. Aquí se mide el número de elementos de datos utilizados por las consultas respondidas por cada uno de los broadcast difundidos por el servidor LBS sobre el número de elementos de datos que son requeridos por las consultas.

Para representar las efectividad de los broadcast también se evalúa las consultas que son creadas en el momento en que se comienzan a difundir los elementos de datos. Por otro lado, también se evalúa el número de elementos de datos que son utilizados por las consultas; sin embargo, a diferencia de la utilidad, este valor se estima sobre el número de elementos de datos que requieren las consultas que fueron creadas.

Entonces, por un lado la utilidad mide el número de elementos de datos que fueron aprovechados por los usuarios, mientras que la efectividad mide el nivel de validez con que un broadcast responde consultas.

### 5.3.1. Algoritmo probabilista con tiempo de espera cambiando los ritmos de schedule

A continuación se presentan los **Cuadro 5.10**, **5.11** y **5.12** que muestran la **utilidad de los broadcast** para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.



Datos (RDU)			
Técnica	<i>elementos encontrados</i>	<i>largo</i>	<i>utilidad</i>
ExW-C	164.1538	483.8462	0.4522
PxW-C	175.6154	480.6154	0.4564
RxW-C	185.9167	523.0000	0.5035

**Cuadro 5.10:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

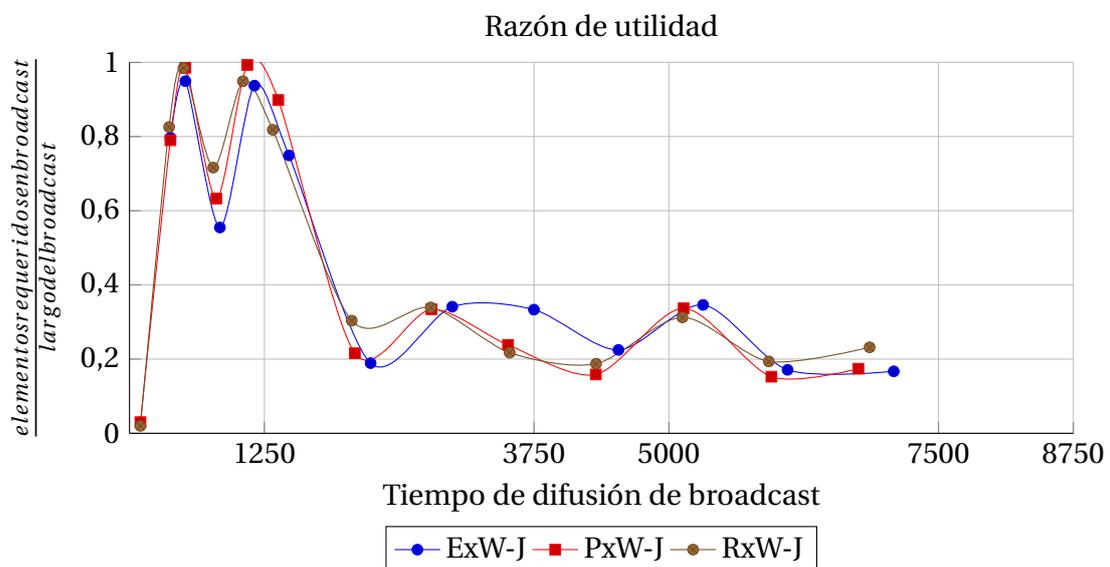
RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

El anterior **Cuadro 5.10** de la utilidad de los broadcast muestra los algoritmos de envergadura, de popularidad y relevancia probabilista con tiempo de espera trabajando bajo el algoritmo de selección de certeza.

Como se puede observar, las tres técnicas muestran pocas diferencias, pensando en que los peak de utilidad se ubican entre el tiempo 0 y 1250. Dado que después del tiempo 1250

los usuarios se comienzan a mover hacia el centro del plano y por otro lado, los usuarios comienzan a cumplir con su demanda predefinida, comienza a bajar la utilidad de los broadcast. En la tabla del gráfico se observa que la utilidad promedio de los broadcast se encuentra entre 45% y 50%, en general los broadcast presenta similares resultados respecto a los elementos encontrados promedio, y el largo promedio de los broadcast.

Luego se observa que para el siguiente **Cuadro 5.11** en donde se agrega la selección por jitter, no existen cambios evidentes en la utilidad. En en siguiente y el anterior se puede observar que los últimos broadcast se tienden a estabilizar en 20% de utilidad.



Datos (RDU)			
Técnica	<i>elementos encontrados</i>	<i>largo</i>	<i>utilidad</i>
ExW-J	189.3333	528.4167	0.4800
PxW-J	163.5385	477.0769	0.4570
RxW-J	170.4615	480.6154	0.4692

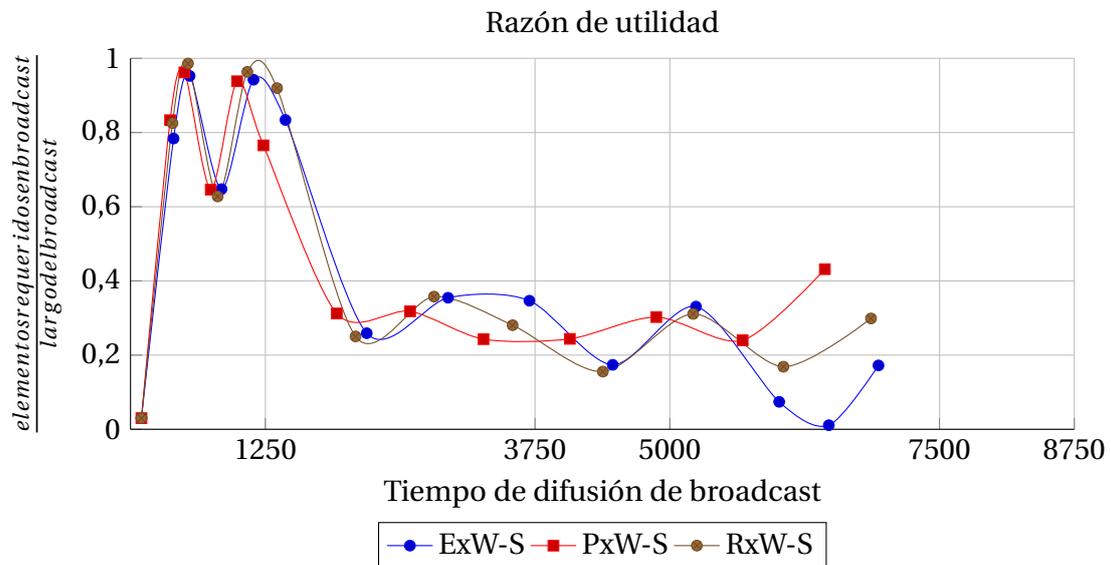
**Cuadro 5.11:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

Finalmente, cuando se cambia el algoritmo de selección al de stretch **Cuadro 5.12** se muestra que la estabilidad de la utilidad en los últimos broadcast es dispareja. El algoritmo de popularidad muestra que su utilidad supera el 40% , mientras que la técnica de envergadura, una vez que se fue a la baja, comienzan regresar al 20% de utilidad, por otro lado, la técnica de relevancia comienza a volver a la curva que tiene desde el tiempo 1250.



Datos (RDU)			
Técnica	$\overline{\text{elementos encontrados}}$	$\overline{\text{largo}}$	$\overline{\text{utilidad}}$
ExW-S	159.1429	483.6429	0.4223
PxW-S	169.0000	454.6154	0.4820
RxW-S	172.9231	484.7692	0.4750

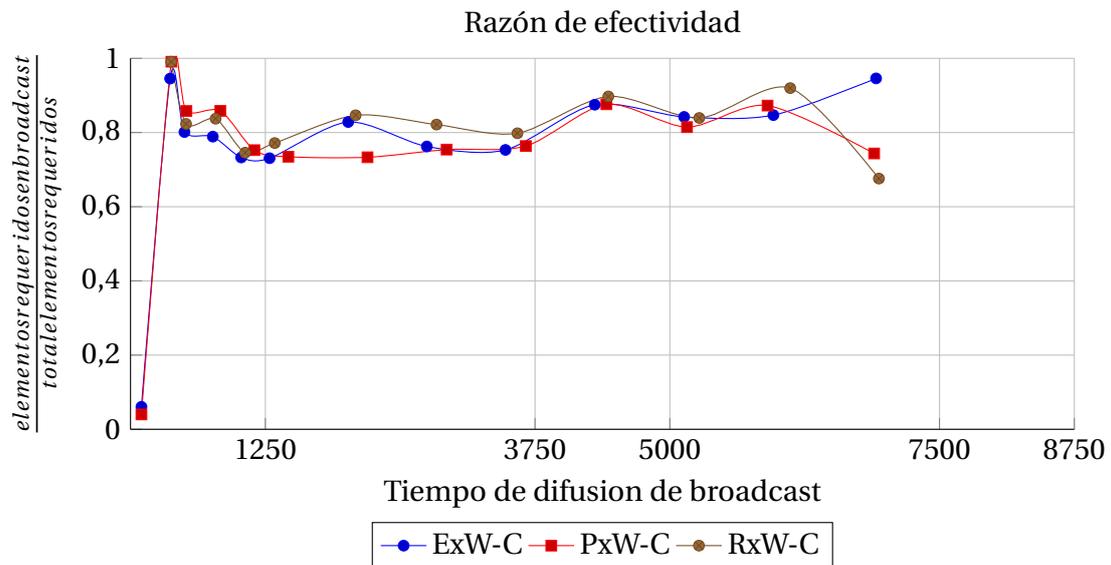
**Cuadro 5.12:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

A continuación se presentan los **Cuadro 5.13**, **5.14** y **5.15** que representan la efectividad de los broadcast para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.



Datos (RDE)			
Técnica	<i>elementos encontrados</i>	<i>elementos requeridos</i>	<i>efectividad</i>
ExW-C	164.1538	206.9231	0.7626
PxW-C	175.6154	222.1538	0.7533
RxW-C	185.9167	226.6667	0.8306

**Cuadro 5.13:**

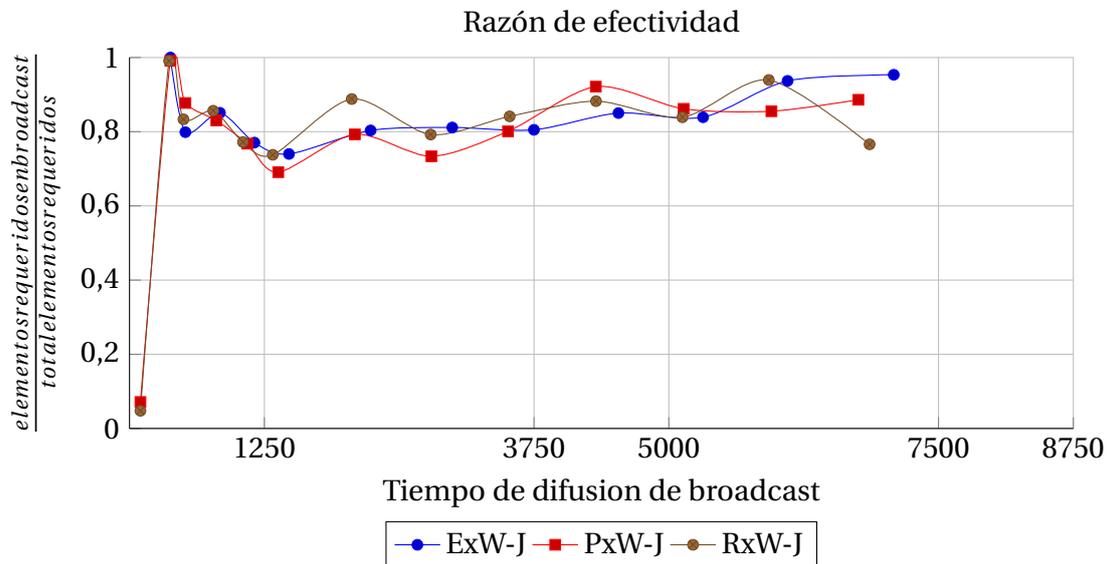
ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]  
 PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]  
 RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

El anterior **Cuadro 5.13** de la efectividad de los broadcast muestra los algoritmos de envergadura, de popularidad y relevancia probabilista con tiempo de espera trabajando bajo el algoritmo de selección de certeza.

Como se puede observar, la efectividad de los broadcast se tiende a acercarse al 80%, lo que significa que los broadcast contienen ese porcentaje de elementos de datos que son requeridos por las consultas. Sin embargo, al final, tanto la técnica de popularidad como de envergadura caen por debajo del 80%. Éste fenómeno se explica por el hecho de que se mide la efectividad de las consultas creadas antes de realizar una difusión, entonces, puede ocurrir que en la última difusión se hayan transmitido elementos que fueron solicitados con anterioridad, sin embargo, antes de comenzar la difusión se crearon pocos elementos de datos, que de todas formas son respondidos por el broadcast.

Tanto la baja como la subida de la efectividad de los broadcast, al final de la simulación, no son indicadores de mayor o mejor rendimiento. Por lo tanto, para medir el rendimien-

to se debe observar el ciclo completo, en donde los algoritmos de scheduling mantienen cortas diferencias, donde en algún momento es mejor la relevancia pero también mejora la popularidad y la envergadura.



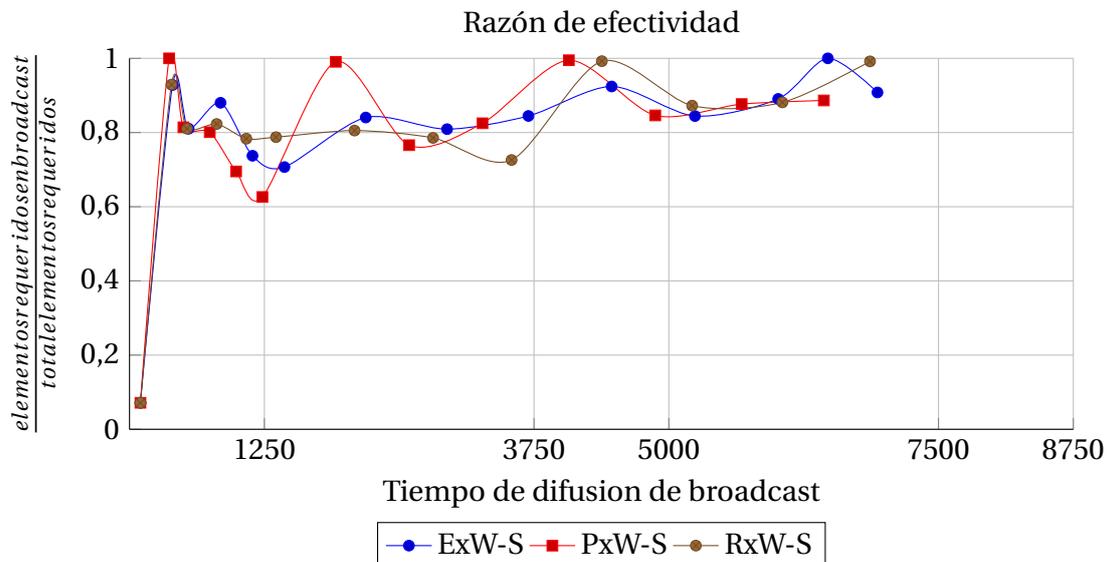
Datos (RDE)			
Tecnica	<i>elementos encontrados</i>	<i>elementos requeridos</i>	<i>efectividad</i>
ExW-J	189.3333	230.0833	0.8467
PxW-J	163.5385	205.8462	0.7754
RxW-J	170.4615	208.2308	0.7836

**Cuadro 5.14:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]



Datos (RDE)			
Técnica	<i>elementos encontrados</i>	<i>elementos requeridos</i>	<i>efectividad</i>
ExW-S	159.1429	196.9286	0.7997
PxW-S	169.0000	208.5385	0.7841
RxW-S	172.9231	212.1538	0.7892

**Cuadro 5.15:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]  
 PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]  
 RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

En conclusión, al variar las técnicas de planificación y los respectivos algoritmos de selección, no se observan mayores cambios en la utilidad y efectividad. Por lo tanto, las técnicas se caracterizan de forma innata por la optimización tanto como de la utilidad como de la efectividad de los broadcast. Esto se puede ver como una ventaja ya que cualquiera de las técnicas podría ser usada, independiente del ambiente en que estén operando, y se obtendrán resultados similares en utilidad y efectividad.

## 5.4. Justicia

La justicia se define como el nivel de justicia de los broadcast respecto en su jitter y su stretch. A diferencia de las técnicas de selección basadas en jitter y stretch, cuando se mide el nivel de justicia del broadcast, se calcula el valor en función de todo los elementos de datos del broadcast; en los algoritmos de selección se mide únicamente los elementos de datos del sub-broadcast de tamaño predefinido.

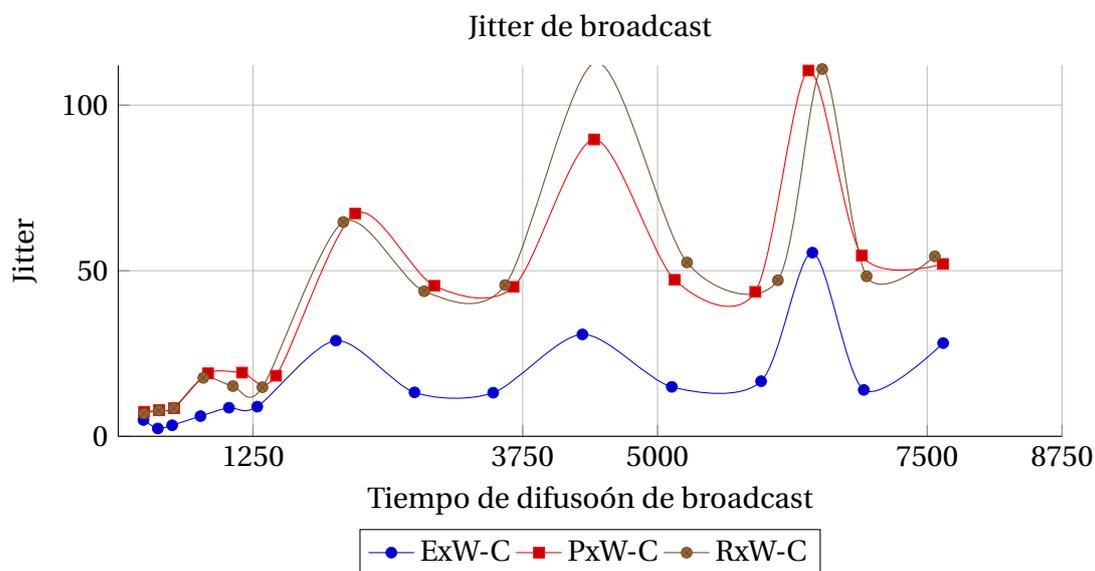
El stretch de un broadcast se mide como la razón entre el tiempo de servicio sobre el número de elementos de datos de una consulta que es respondida por un broadcast. Esto significa que un broadcast mas justo en función del stretch establece que las consultas tienen un tiempo de servicio en función del número de elementos de datos que requieren, es decir, es más justo que las consultas que requieren menos elementos esperen menos tiempo que las consultas que requieren más elementos.

Por otro lado, el jitter de un broadcast se mide como la distancia entre los elementos de datos que responden una consulta en el broadcast. Esto significa que un broadcast mas justo en función del jitter envía los elementos de datos que responden una consulta separados con una distancia ecuánime, es decir, es más justo que las consultas que se responden dentro de un broadcast reciban los elementos de datos en periodos similares de tiempo.

Para calcular, el jitter de un broadcast se calcula la distancia de cada una de las consultas que fueron respondidas y a ellas se les calcula el promedio. Por otro lado para el stretch se realiza la misma operación, es decir, se calcula el stretch de cada una de las consultas que son respondidas en un broadcast y con ellas se calcula el promedio.

### 5.4.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule

A continuación se presentan los **Cuadro 5.16 ,5.17 y 5.18** con los parámetros de justicia desde el punto de vista del jitter de los broadcast para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.



Datos jitter			
Técnica	Máximo jitter	Mínimo jitter	$\overline{jitter}$
ExW-C	55.4565632319	2.3680292103	16.6404
PxW-C	110.444272552	7.41045426065	42.3888
RxW-C	112.919789626	6.86395989975	43.4063

**Cuadro 5.16:**

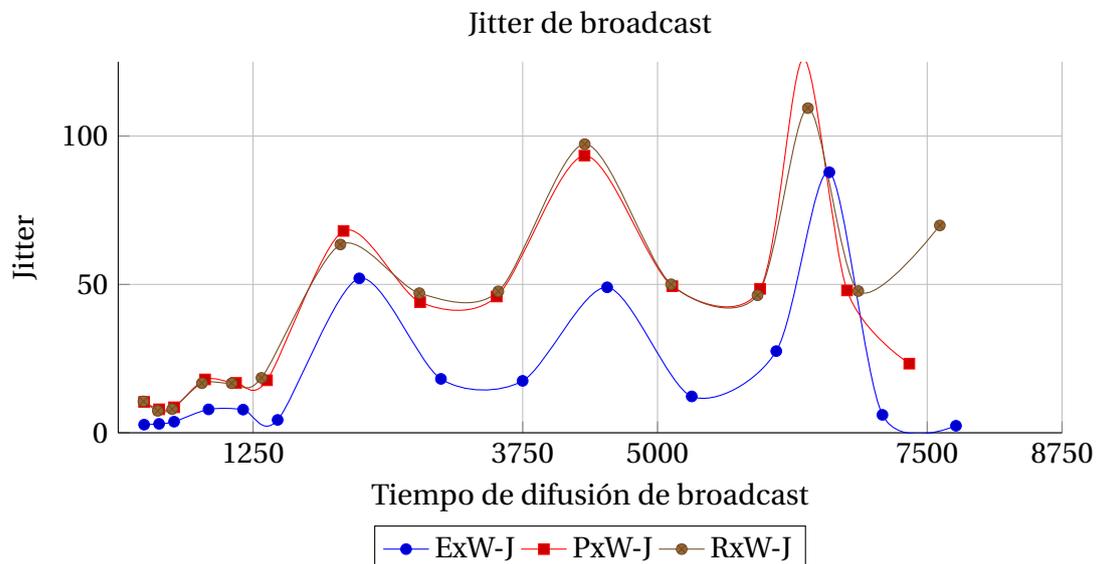
ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Como se puede observar en el anterior **Cuadro 5.16** el algoritmo de envergadura es el que muestra los peak de jitter mas bajos. El hecho de que presente una curva mas estable a través del tiempo es porque la técnica ordena los elementos de datos por conjuntos, esto significa que a la hora de calcular el jitter, los elementos de datos están separados por menores distancias. Por otro lado, las técnicas de scheduling de popularidad y relevancia muestran una curva que después del tiempo 3750 sobrepasa con creces el limite de los 50 puntos.

No obstante, el hecho de que una u otra técnica tenga mayores o menores valores de jitter, no tiene una mayor importancia en la justicia. Es por ello que la justicia se debe observar como la estabilidad de la curvatura. Donde en el mejor de los casos, se establecería una línea horizontal, de modo que, sin importar el escenario, siempre se mantuviera el mismo nivel de distancia de los elementos de datos que son respondidos en un broadcast.

Por otro lado, en el **Cuadro 5.17** mostrado a continuación, se observa que al aplicar la selección por jitter, que si bien se aplica al sub-broadcast, la brecha que separa a las tres

diferentes técnicas, se acorta de forma considerable. Esto significa que la única forma de hacer ecuánime los broadcast con respecto al jitter es aplicar la selección por jitter.

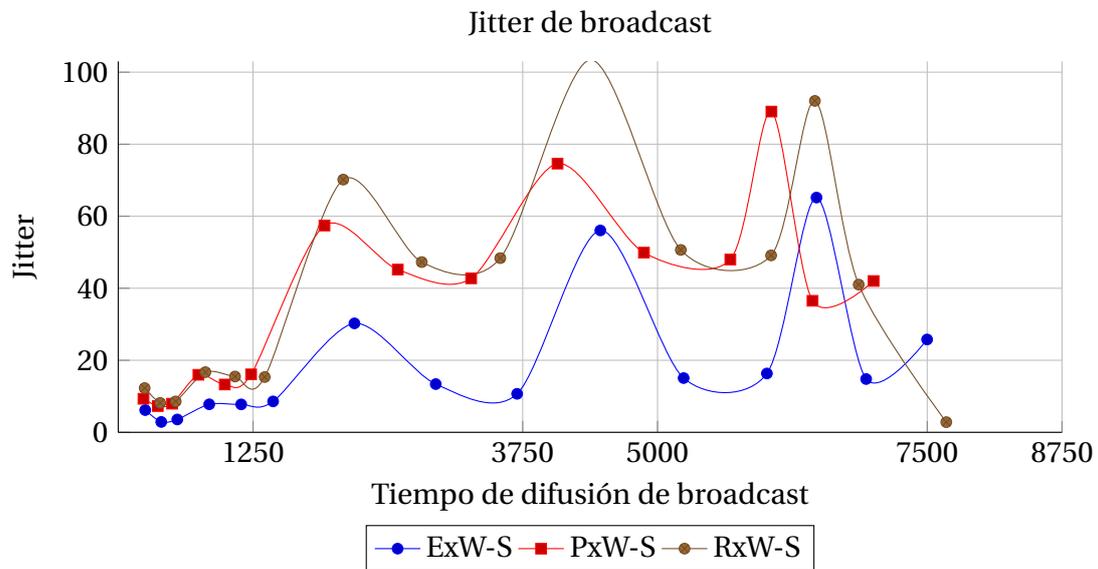


Datos jitter			
Técnica	Máximo jitter	Mínimo jitter	$\overline{jitter}$
ExW-J	87.7820064454	2.36978021978	20.1481
PxW-J	125.966292081	7.91765338033	41.7271
RxW-J	109.394828342	7.39089541582	43.7799

**Cuadro 5.17:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Finalmente, en el **Cuadro 5.18** mostrado a continuación se observa un resultado con aparente injusticia respecto al jitter. Esto se debe a que las curvas no se estabilizan, es decir, no se observan peaks similares a través del tiempo.



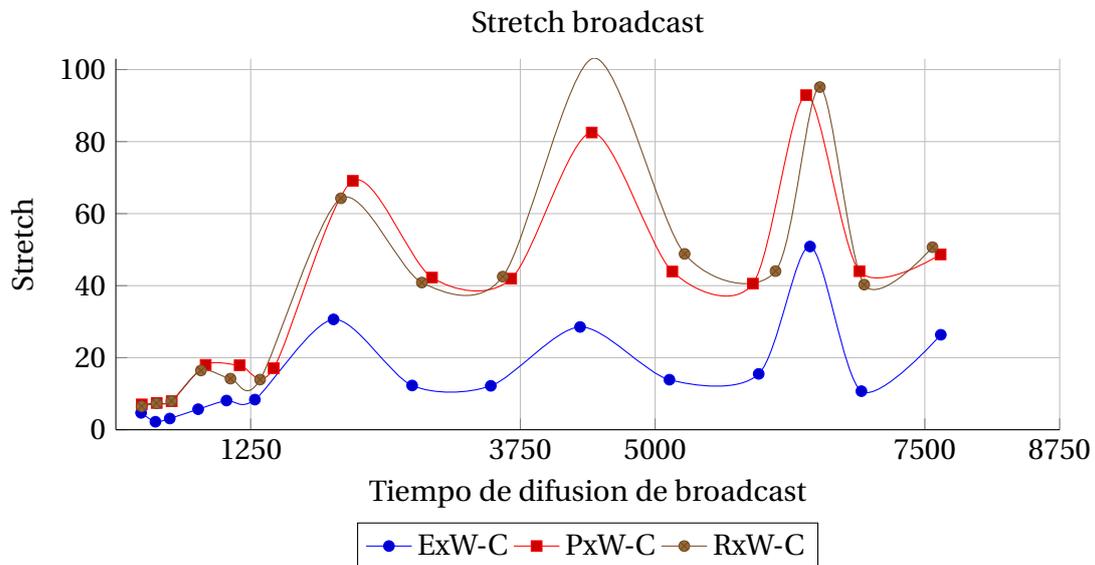
Datos jitter			
Técnica	Máximo jitter	Mínimo jitter	$\overline{jitter}$
ExW-S	65.1868382609	2.88152264629	18.9517
PxW-S	89.0605365996	7.29302136134	37.0221
RxW-S	103.417928889	2.82352941176	38.7619

**Cuadro 5.18:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

En conclusión, tanto la técnica de selección por certeza como por jitter tienden a estabilizar los peaks. Por otro lado, La selección por jitter, a pesar de que estabiliza las curvas, tiende a aumentar los valores en la técnica de scheduling de envergadura. A favor de la justicia del jitter, este último fenómeno se observa como el mejor de los escenarios, ya que no se cambia el tiempo de respuesta, a pesar de que los usuarios comienzan a recibir los elementos de datos en lapsos muy similares.

A continuación se presentan los **Cuadro 5.19 , 5.20 y 5.21** con los parámetros de justicia desde el punto de vista del stretch de los broadcast para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.



Datos Stretch			
Técnica	Máximo stretch	Mínimo stretch	$\overline{stretch}$
ExW-C	50.8742473563	2.21969086388	15.5440
PxW-C	92.9081661764	7.03520647773	38.7536
RxW-C	103.175211565	6.51728744939	39.7425

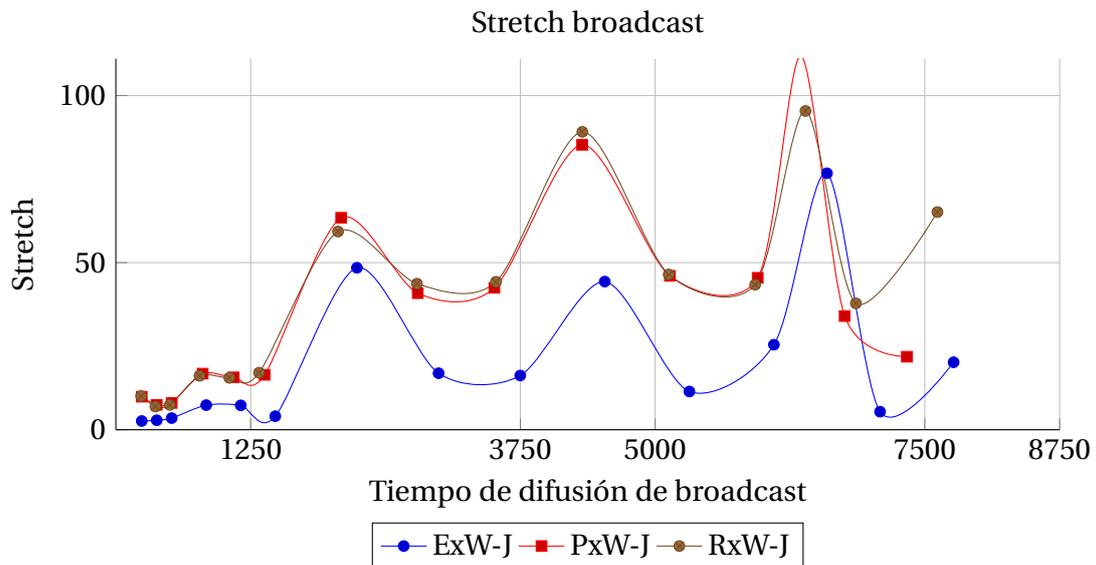
**Cuadro 5.19:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Como se puede observar el anterior **Cuadro 5.19**, el nivel de stretch es muy similar entre la técnica de popularidad y relevancia, sin embargo los valores de la técnica de envergadura son valores evidentemente menores. Esto significa que las consultas son respondidas en lapsos de tiempo más cortos y el hecho de que la curvatura sea mas o menos estable significa que las consultas se están respondiendo en proporción a su tamaño.

Cuando observamos el stretch promedio en la tabla de valores del **Cuadro 5.19**, se puede notar que los puntos de la técnica de envergadura son aquellos que se acercan en mayor medida a los puntos de las otras técnicas; ésta es otra forma de observa la estabilidad del stretch a través del tiempo.

Al igual que los anteriores **Cuadro 5.16**, **5.17** y **5.18** que representan el jitter, se observa que la técnica de selección por jitter estabiliza el stretch de los broadcast, aumentando los niveles de stretch de la técnica de envergadura. Mientras que la técnica de selección de stretch tiene mayores variaciones a través del tiempo.



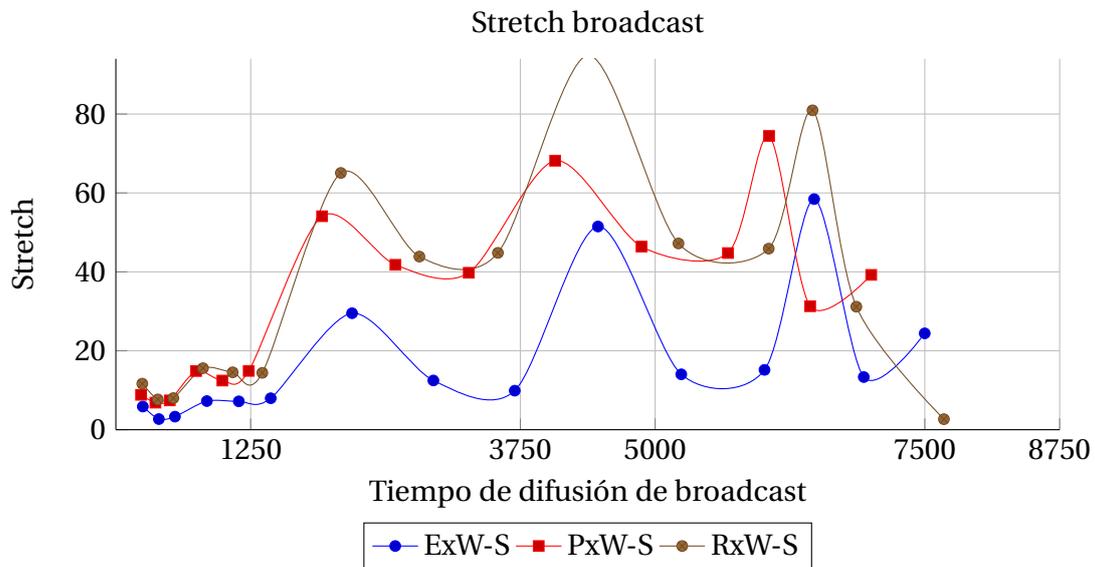
Datos Stretch			
Técnica	Máximo stretch	Mínimo stretch	$\overline{stretch}$
ExW-J	76.737725902	2.600682861	19.5132
PxW-J	111.766331266	7.44030613794	37.6871
RxW-J	95.3900694413	6.94829658682	39.8488

**Cuadro 5.20:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]



Datos Stretch			
Técnica	Máximo stretch	Mínimo stretch	$\overline{stretch}$
ExW-S	58.4348524596	2.69437118907	17.5366
PxW-S	74.4492413997	6.85598141186	33.6867
RxW-S	94.8739792944	2.66666666667	35.2240

**Cuadro 5.21:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

En conclusión, cuando se aplica la técnica de selección por certeza se estabiliza la técnica de envergadura, tanto en la justicia de jitter como de stretch, mientras que la técnica de selección por jitter acorta la brecha entre las diferentes técnicas de scheduling, tanto en la justicia del jitter como en stretch. Sin embargo, la técnica de selección por stretch, tiende a desestabilizar la justicia tanto del jitter como del stretch de los broadcast.

Lo anterior significa que en circunstancia cuando se requiere responder de forma justa las consultas, es preferible aplicar la técnica de selección por jitter. Por ejemplo, cuando comienza a aumentar la redundancia espacial y temporal en un sector de importantes dimensiones en el plano.

## 5.5. Obsolescencia

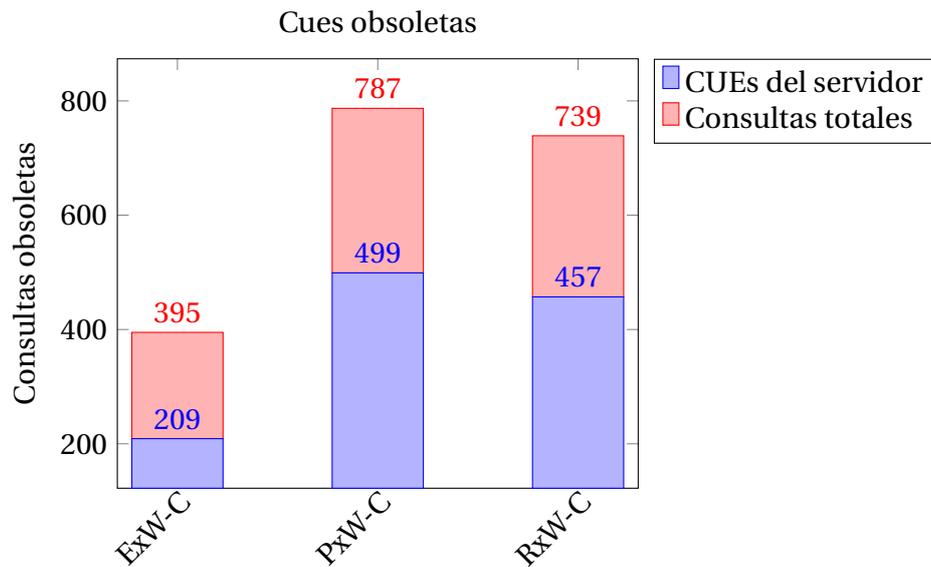
La obsolescencia es una medida del rendimiento del servidor LBS, donde se analiza el número de CUEs caducadas para cada una de las técnicas de scheduling. La obsolescencia se mide desde que el usuario envía su consulta al servidor LBS hasta que el servidor LBS completa su consulta exacta. De este modo que, cuando un usuario se sale de la región de encubrimiento de su CUE ésta queda obsoleta.

Para mostrar la obsolescencia de las simulaciones se separan las consultas obsoletas que fueron enviadas al servidor y las consultas obsoletas que no llegaron al servidor LBS. Como se puede pensar, las consultas que no son enviadas al servidor no deberían quedar obsoletas, sin embargo, pensando en que los usuarios deben completar una consulta antes de crear una nueva, cuando los usuarios observan un broadcast antes de enviar su consulta, esa consulta para la que están buscando los elementos de datos, también puede quedar obsoleta, ya que el usuario se está moviendo a través del tiempo.

La obsolescencia medida en este trabajo es estricta, debido a que se observa únicamente que el usuario esté dentro de la región de encubrimiento de su consulta de rango. por otro lado, es una medida que solo puede calcular el usuario, ya que el servidor no sabe cual de todas las cultas de rango es la que pertenece al usuario y, por ende, tampoco conoce la posición del usuario.

### 5.5.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule

A continuación se presentan los **Cuadro 5.22, 5.23 y 5.24** con los parámetros de obsolescencia de los broadcast para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.



Datos obsolescencia		
Técnica	Obsoletas en el servidor	Obsoletas en el broadcast
ExW-C	209	186
PxW-C	499	288
ARPT-C	457	282

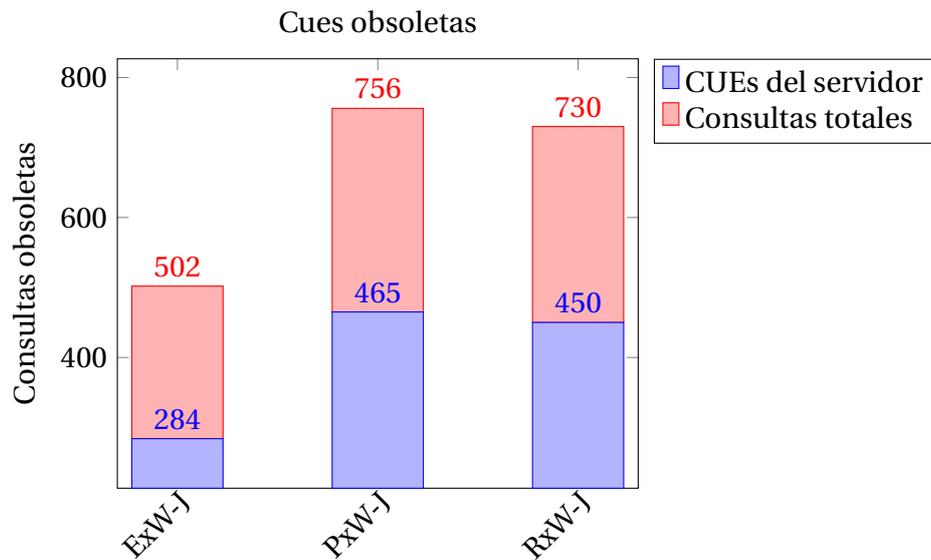
**Cuadro 5.22:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]  
 RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Como se puede observar en el **Cuadro 5.22**, las técnicas de scheduling de popularidad y relevancia presentan mayores niveles de obsolescencia, tanto en las consultas que llegaron al servidor como las que no llegaron al servidor.

El anterior **Cuadro 5.22** el comportamiento se debe a que la técnica de schedule de envergadura agrupa los elementos de datos según las consultas que se deben responder. Esto significa que los elementos de datos requeridos por las consultas, con la técnica de envergadura, siempre se van a obtener más rápido que las otras técnicas. y pensando en la estricta condición de obsolescencia que aquí se plantea, mientras más rápido se responde, menos consultas quedaran obsoletas en la simulación.

Dado lo anterior, es importante observar el comportamiento de las técnicas de popularidad y relevancia **Cuadro 5.23**. Donde se observa que la técnica de popularidad tiene mayores índices de obsolescencia; tanto en el gráfico anterior como en el que sigue.



Datos obsolescencia		
Técnica	Obsoletas en el servidor	Obsoletas en el broadcast
ExW-J	284	218
PxW-J	465	291
RxW-J	450	280

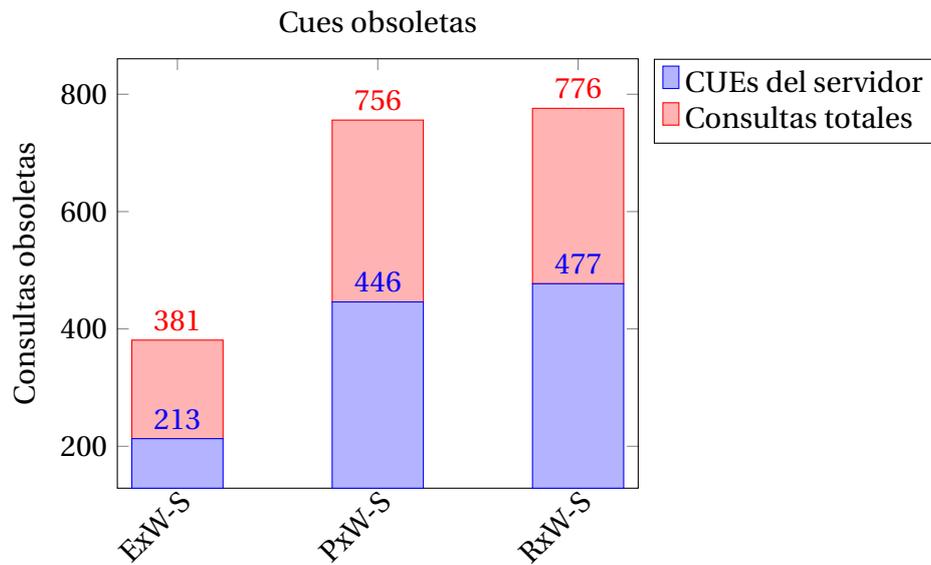
**Cuadro 5.23:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Finalmente el **Cuadro 5.24** mostrado a continuación muestra un ligero cambio en el nivel de obsolescencia, donde, se puede observar que el algoritmo de relevancia, sobrepasa a la popularidad, aunque sea por 20 puntos de obsolescencia.



Datos obsolescencia		
Técnica	Obsoletas en el servidor	Obsoletas en el broadcast
ExW-S	213	168
PxW-S	446	310
RxW-S	477	299

**Cuadro 5.24:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

A modo de conclusión, la técnica de envergadura es la que obtiene menores índices de obsolescencia cuando se aplica la técnica de selección por stretch. Esto se debe principalmente a que los elementos de datos están más cerca entre si cuando se utiliza la técnica de envergadura. Entonces, a menor tiempo de respuesta, siempre será menor el índice de obsolescencia.

## 5.6. Privacidad

La privacidad es una medida del rendimiento del servidor LBS, donde se analiza el número de CUEs que son respondidas sin ser enviadas al servidor LBS. Por otro lado, se plantea una métrica de evaluación de la privacidad como el nivel de la incertidumbre promedio de las CUEs que llegan al servidor. Esta métrica de evaluación de la privacidad se basa en el estudio de Niu (Niu y cols., 2015).

Para evaluar el nivel de incertidumbre promedio de las CUEs, se plantea una variación de la formula presentada por Niu (Niu y cols., 2015). Esta formula permite medir el nivel de privacidad que tienen un servidor LBS según las distintas técnicas de scheduling que se aplican.

$$\lambda = \frac{\sum_{q \in Q} (\log_2(\frac{1}{k_q}) + \log_2(N) \times |Q_{no\ enviadas\ y\ respondidas}|)}{|Q_{enviadas\ y\ respondidas}| + |Q_{no\ enviadas\ y\ respondidas}|}$$

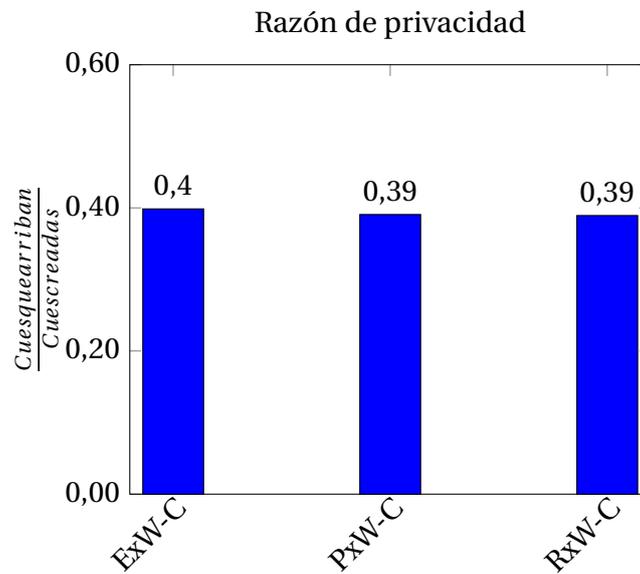
Donde  $N$  corresponde al número de celdas del plano en que los usuarios se mueven y  $k$  corresponde al nivel de anonimato de cada una de las CUEs respondidas.

En este trabajo para evaluación preliminar del nivel de privacidad que otorgan los algoritmos de scheduling en combinación con las técnicas de selección se presenta la razón entre el el número de consultas que arribaron al servidor LBS sobre el número de CUEs que fueron creadas.

### 5.6.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule

A continuación se presentan los **Cuadro 5.25, 5.26 y 5.27** con los parámetros de privacidad de los broadcast para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el  $k$  anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.

En los **Cuadro 5.25, 5.26 y 5.27** mostrados a continuación se observa que el nivel de privacidad es similar entre las diferentes técnicas de schedule.

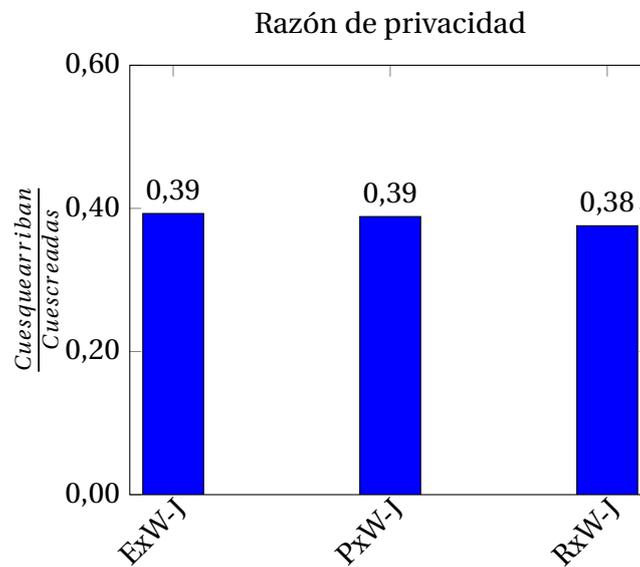


**Cuadro 5.25:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

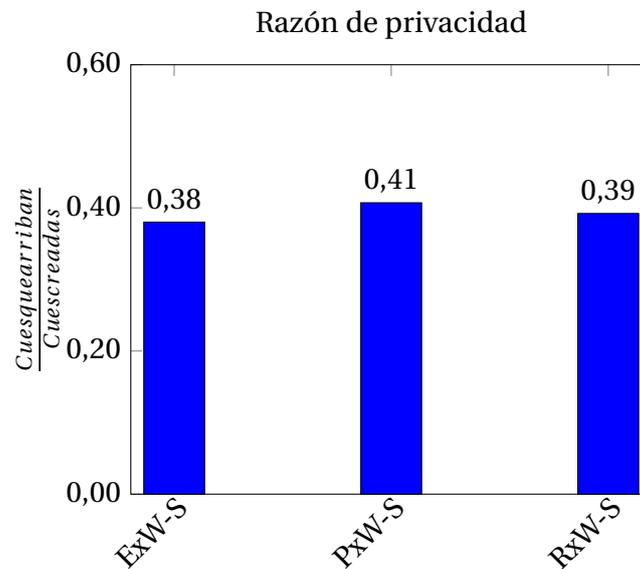


**Cuadro 5.26:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

**Cuadro 5.27:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Efectividad=3, Talla=100]

En conclusión las técnicas de planificación, a pesar de que discrepan en el tiempo de respuesta, su nivel de efectividad al momento de responder consultas, antes de que ellas sean enviadas al servidor LBS, es similar, por lo tanto, bajo esta condición de trabajo el los algoritmos de planificación deben procesar cantidades muy similares de CUEs mientras que están respondiendo de forma efectiva, de igual forma, cantidades muy similares de consultas.

## 5.7. Carga de trabajo

La carga de trabajo del servidor es una medida de rendimiento de los algoritmos de scheduling para evaluar el número de CUEs que responde el servidor LBS. El servidor LBS optimiza su carga de trabajo cuando de las consultas creadas recibe tan pocas como sea posible.

En otras palabras, el hecho de que el servidor reciba menos CUEs significa que los elementos de datos que está transmitiendo esta siendo efectivos para satisfacer las consultas que los usuarios están requiriendo.

Sin embargo, el factor del tiempo también es importante a la hora de ver la carga de trabajo del servidor, ya que, con determinada técnica de scheduling el servidor puede estar recibiendo una baja cantidad de CUEs, pero el tiempo que le toma responder una determinada cantidad de consultas está siendo relativamente mayor.

Dado lo anterior, cuando una técnica de planificación optimiza su carga de trabajo es cuando la razón entre el número de consultas creadas o respondidas por el servidor sobre el número de CUEs que son demandadas al servidor es bajo y además el tiempo en que le toma responder las consultas es bajo.

### 5.7.1. Algoritmo probabilista con tiempo de espera cambiando los algoritmos de schedule

A continuación se presentan los **Cuadro 5.28, 5.29 y 5.30** que representan los valores de la carga de trabajo del servidor y el sistema para los **algoritmos de scheduling de envergadura, popularidad y relevancia probabilista con tiempo de espera** y los algoritmos de selección de certeza donde se mantiene el k anonimato, el número de puntos de interés, el número de repeticiones, el tamaño del broadcast y el número de usuarios.

El **Cuadro 5.28** mostrado a continuación muestra las diferentes técnicas de scheduling con el algoritmo de selección por certeza. Allí se puede observar que CDT (carga de trabajo) es muy similar entre todas las técnicas, sin embargo la que optimiza de mejor manera su carga de trabajo es la técnica de relevancia sin embargo es la que tienen el segundo mayor tiempo de trabajo. En este caso la técnica más equilibrada es la popularidad.

Técnica	Carga trabajo del sistema (CDTs)			
	$ Q_{Creadas} $	$ Q_{Arribadas} $	$T_{trabajo}$	$CDTs$
ExW-C	1400	558	7648	2.5090
PxW-C	1400	547	7646	2.5594
RxW-C	1400	545	7571	2.5688

**Cuadro 5.28:**

ExW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-C->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

En segundo lugar, el **Cuadro 5.29** mostrado a continuación muestra las diferentes técnicas de scheduling con la técnica de selección por jitter. Allí se puede observar nuevamente que la popularidad es la técnica que optimiza de mejor forma la carga de trabajo, ya que llegan pocas consultas y se tarda menor tiempo en completar el trabajo.

Técnica	Carga trabajo del sistema (CDTs)			
	$ Q_{Creadas} $	$ Q_{Arribadas} $	$T_{trabajo}$	$CDTs$
ExW-J	1400	550	7767	2.5455
PxW-J	1400	544	7334	2.5735
RxW-J	1400	526	7617	2.6616

**Cuadro 5.29:**

ExW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-J->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

Finalmente, el **Cuadro 5.30** mostrado a continuación presenta las técnicas de scheduling con el algoritmo de selección por stretch. Allí se puede observar que la envergadura es la técnica que optimiza de mejor forma la carga de trabajo, ya que llegan pocas consultas y se tarda un tiempo aceptable en completar el trabajo.

Técnica	Carga trabajo del sistema (CDTs)			
	$ Q_{Creadas} $	$ Q_{Arribadas} $	$T_{trabajo}$	$CDTs$
ExW-S	1400	532	7499	2.6316
PxW-S	1400	570	7003	2.4561
RxW-S	1400	549	7677	2.5501

**Cuadro 5.30:**

ExW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

PxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

RxW-S->[Usuarios=200, K-anonimato=4, POIs=1600, CUEs=1400, Repeticiones=3, Talla=100]

En conclusión, para las técnicas de scheduling de popularidad presentan los menores tiempos para completar el trabajo de responder una determinada cantidad de consultas. sin embargo no siempre recibe menos CUEs. Sin embargo, para cada una de las técnicas el rango de CUEs que llegan al servidor siempre se encuentra entre las 525 y 570, por lo tanto la brecha no para ser muy significativa.

## Capítulo 6

### Conclusiones y trabajos futuros

En esta tesis se crea y evalúa el rendimiento de tres técnicas de scheduling tales como envergadura, popularidad y relevancia, las cuales generan un broadcasting que es ajustados con técnicas como certeza, jitter y stretch. Estas técnicas en su conjunto equilibran las siguientes métricas de rendimiento: carga de trabajo del servidor LBS, la obsolescencia de las respuestas del broadcast, la latencia que experimenta cada usuario mientras espera por los datos que responden su CUE, la injusticia en los tiempos de respuesta obtenidos cuando un grupo de usuarios espera por sus respuestas en el broadcast y la pérdida de la privacidad de la ubicación del usuario.

En la literatura, si bien hay técnicas que equilibran u optimizan las anteriores métricas, no existe un enfoque en el procesamiento de consultas con ubicación encubierta, en las cuales, se observan datos irrelevantes que serán puestos en el broadcasting; debido a que los usuarios ocultan la ubicación de su consulta espacial con  $k-1$  consultas similares, pero en otras ubicaciones. Por otro lado, en la literatura se observa un amplio debate para demostrar la importancia de optimizar la obsolescencia o el tiempo de respuesta, sin embargo, en este trabajo se considera que con la técnicas de scheduling presentadas es posible equilibrar ambas.

La técnica de envergadura tiene como objetivo equilibrar la latencia que experimenta cada usuario mientras espera por las respuesta que responden su CUE. Esto se aplica debido a que los usuarios tienen la posibilidad de escoger su nivel de anonimato y mientras sea más alto aumenta el tamaño de las CUEs. No obstante, en la literatura se asume que si una consulta requiere más elementos de datos esta tiene mayor utilidad ya que ayuda responder otras.

La técnica de envergadura tiene un rendimiento mejor que las otras técnicas en el tiempo de respuesta promedio; esto se observa en el capítulo de evaluación del sistema. Esta técnica

al priorizar las CUEs que solicitan pocas respuestas, garantiza que un mayor número de usuarios tendrá su respuesta en lapsos de tiempo más cortos. Esto trae como consecuencia que métricas como la utilidad, efectividad, justicia y obsolescencia también se equilibren, por lo tanto, la métrica de envergadura tienen una relación de proporción directa con las otras.

La técnica de popularidad tiene como objetivo responder en un menor número de buckets a la mayor cantidad de usuarios que se concentran en una determinada zona, de modo que se equilibre la latencia que experimenta cada usuario mientras espera por las respuesta que responden su CUE y se equilibre la obsolescencia de las CUEs.

La técnica de relevancia tiene como objetivo responder en un menor número de buckets a la mayor cantidad de usuarios que utilizan un bajo nivel anonimato y se concentran en una determinada zona, de modo que se equilibre la latencia que experimenta cada usuario mientras espera por las respuesta que responden su CUE y se equilibre la obsolescencia de las CUEs.

Como se observa en el capítulo estrategias de scheduling propuestas, la técnica de popularidad y relevancia son parecidas, con la diferencia de que en la popularidad cada respuesta tiene un nivel de prioridad asociada con el número de CUEs que la requieren, sin embargo, en la relevancia las respuestas tienen un nivel de prioridad asociado al número de CUEs que la solicitan y además observa que las CUEs que la solicitan tengan un bajo nivel de anonimato. Estas dos últimas técnicas presentadas tienen un menor rendimiento que la técnica de envergadura respecto a la latencia que experimenta cada usuario. Sin embargo, se debe considerar que estas técnicas presentan mejores resultados con respecto a la pérdida de privacidad. En otras palabras, al mejorar la latencia de los usuarios, se pierde la privacidad ya que deben enviar más CUEs al servidor LBS para que este pueda responder. Esto es debido a que con las técnicas de popularidad y relevancia los usuarios tienen un menor tiempo de procesamiento. El tiempo de procesamiento es el transcurrido entre el momento en que los usuarios crean su CUE hasta que reciben la primera respuesta; considerando que el usuario primero observa un broadcast, para saber si sus respuestas están siendo transmitidas antes de enviar una CUE al servidor LBS.

Entonces, en este trabajo se han alcanzado los objetivos planteados y se han obtenido resultados sobre el comportamiento de algoritmos de scheduling, los cuales aportan en gran medida a que el servidor LBS realice su trabajo de forma más equitativa, de modo que los usuarios responden una gran cantidad de consultas sin que estas sean enviadas y, por lo tanto, equilibran su privacidad de ubicación.

Por otro lado, las técnicas de ajustes presentadas, tiene la finalidad de ecuanimizar el

broadcasting. Esto significa que, dentro de una serie de posibles broadcast, con los algoritmos de ajuste, se selecciona aquel que presente mejores resultados respecto a dichas técnicas. Sin embargo, en las pruebas se observa que estas técnicas de ajuste no tiene mayor influencia en los resultados finales.

Es importante destacar la implementación del simulador que permite crear distintos ambientes de prueba, tanto es así, que es posible modificar desde al nivel de anonimato de las CUEs hasta el tamaño de las regiones de encubrimiento, pasando también por los patrones de movimiento de los usuarios.

## 6.1. Recapitulación

En la introducción se presenta los conceptos mas generales de este trabajo. Se muestran los principales asuntos motivadores como la privacidad de la ubicación de los usuarios móviles de un sistema LBS. Por otro lado, se sientan las bases para entender la dirección de este trabajo, en donde, se plantea, mediante diferentes técnicas de scheduling, construir sistemas de broadcast eficientes que equilibran métricas como el tiempo de respuesta, la utilidad y efectividad, la obsolescencia, la justicia, la privacidad y la carga de trabajo del servidor LBS.

En el estado del arte se demuestra que este trabajo sienta sus bases en sistemas de scheduling que difunden información a través de canales aéreos (broadcasting). Esta área a evolucionado en al tiempo debido a la incorporación de nuevas tecnologías y el aumento de información que generan tanto los sistemas como los usuarios. Por otra parte se presentan diversas técnicas de planificación, las cuales son construidas para diferentes ambientes. De la literatura se observa que, si bien existen diferentes técnicas de schedulin, esta abierta a investigación en el área de la planificación de respuestas para usuarios móviles que quieren resguardar tanto como sea posible su ubicación y de ese modo obtener anonimato.

En los alcances de la investigación se descartan aspectos como la arquitectura del sistema, la cual esta abierta a ser modificada para tener utilidad en diferentes ambientes para resolver consultas con la ubicación encubierta. Por otro lado, se hace una introducción a los aspectos mas importantes del ambiente en que se realiza la planificación de CUEs.

En las estrategias de scheduling se presentan las tres técnicas que son utilizadas en este trabajo para planificar CUEs, las cuales responden de diferente modo dependiendo del comportamiento de los usuarios de un LBS. Lo mas destacado de los diferentes algoritmos es que son adaptaciones que fueron probadas en otros trabajos de la literatura. En otras palabras, se mantiene la idea general para luego ir puliendo los detalles. Por otro lado, las

técnicas son relativamente fáciles de entender e implementar ya que están diseñadas bajo los mismos términos.

Luego se presentan los algoritmos para la selección de broadcast y la técnica de ajuste. Estas últimas son aplicadas porque los algoritmos de scheduling presentados, si se repiten varias veces, presentarían distintos broadcast, y por eso se debe seleccionar alguno que cuente con los mayores índices ya sea de certeza, jitter o stretch.

Con respecto a la hipótesis se confirma que las técnicas de scheduling que construyen broadcast de difusión optimizan tanto como les es posible las métricas de rendimiento. Por otro lado se confirma sobre la posibilidad de construir algoritmos de scheduling que satisfacen tanto al usuario como al servidor LBS. Por otro lado, se confirma que mientras más pro-activo es un servidor LBS se mejora la privacidad de los usuarios. En otras palabras, con la construcción de algoritmos de predicción de respuesta, es posible que los usuarios no deban enviar CUEs a un servidor LBS.

Finalmente, se completan tanto el objetivo general como los objetivos específicos de esta investigación. En primer lugar, se presentan tres diferentes métricas de scheduling; envergadura, popularidad y relevancia. Las cuales equilibran la carga de trabajo, la obsolescencia, la injusticia y la privacidad, pero se debe dejar claro que la técnica de envergadura optimiza mejor el tiempo de respuesta. En segundo lugar se realiza un estudio bibliográfico con el objetivo de sustentar tanto las técnicas de scheduling como la forma en que se evalúa el broadcast. Luego, se propone una nueva métrica de privacidad que se basa en cómo los usuarios pierden la privacidad al enviar consultas al servidor LBS. Finalmente, se genera un simulador con la capacidad de ejecutar un gran número de pruebas sobre los algoritmos de scheduling; se debe destacar que en el presente trabajo se muestran únicamente 3 diferentes variedades de resultados de 238 obtenidos.

## 6.2. Trabajos futuros

Los trabajos propuestos se presentan sobre tres ejes: incorporar un algoritmo que evalúe la efectividad en función de la demanda de CUEs para repetir el broadcast, incorporar índices aéreos de modo que los usuarios puedan saber si los elementos de datos están en el broadcast para no esperar una difusión completa, incorporar un ajuste temporal que considere el caso en que los usuarios están repartidos en el plano de modo que se solicitan todos los elementos de datos del sistema, incorporar una meta heurística que permita identificar la técnica de scheduling que se debe aplicar según el comportamiento de la demanda y el movimiento de los usuarios e incorporar un detector del patrón de movimiento de los usuarios de modo que

el servidor responda una CUEs en fusión de la futura posición del usuario.

En este trabajo se asume que la efectividad del broadcast es una constante que permite repetir una determinada cantidad de veces el broadcast. Esto se debe a que se asume que los usuarios tienen una determinada redundancia en el espacio a través del tiempo. Por lo tanto, es muy probable que un broadcast construido en un tiempo  $t$ , sea efectivo en un tiempo  $t+1$ . Sin embargo, parece ser que la métrica de la efectividad es un valor que varía a través del tiempo, y bajo ese supuesto, es posible realizar diversos análisis del comportamiento de un sistema LBS de modo que se plantee la métrica de efectividad como una variable que debe ser modificada en función de los patrones de movimiento y solicitudes de los usuarios.

En la literatura se observa que los índices aéreos ayudan a que el usuario tenga un rendimiento óptimo en el transcurso que espera sus consultas, de modo que puede tener un ahorro significativo de la energía y en el caso en que se busca proteger la privacidad del usuario, es una gran ventaja que este último siempre esté al tanto de los elementos de datos que se difunden por un servidor LBS.

Generar distintas soluciones para planificar un broadcast y además generar distintos algoritmos de ajuste, da el pie para considerar una meta heurística que permita evaluar en tiempo real cual es la combinación de técnicas más idónea para planificar un broadcast.

En esta investigación el comportamiento de los usuarios se enfoca en esperar los elementos de datos requeridos por su CUE y se mueve a través de un plano de dos dimensiones. Sin embargo, mediante la construcción del simulador es posible notar que el usuario perfectamente puede utilizar la información de su caché para múltiples propósitos. Entre ellos está la idea de que el usuario mantenga los datos históricos para responder sus CUEs con los datos recibidos en otras difusiones de broadcast.

Finalmente, si bien en esta investigación se modelan las técnicas de scheduling considerando que los usuarios tienen diferentes comportamientos de movimiento, no se considera el hecho de que existe la posibilidad de pronosticar las futuras posiciones y por ende crear broadcast pensando en que las CUES deben ser desplazadas según dicha predicción de movimiento.

# Referencias

- Acharya, S., Franklin, M., y Zdonik, S. (1995). Dissemination-based data delivery using broadcast disks. *IEEE Personal Communications*, 2(6), 50–60.
- Acharya, S., y Muthukrishnan, S. (1998). Scheduling on-demand broadcasts: New metrics and algorithms. En *Proceedings of the 4th annual acm/ieee international conference on mobile computing and networking* (pp. 43–54).
- Alabdulatif, A., Kumarage, H., Khalil, I., y Yi, X. (2017). Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption. *Journal of Computer and System Sciences*, 90, 28–45.
- Alizadeh-Shabdiz, F., Jones, R. K., Morgan, E. J., y Shean, M. G. (2007, diciembre 4). *Location-based services that choose location algorithms based on number of detected access points within range of user device*. Google Patents. (US Patent 7,305,245)
- Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., y Palamidessi, C. (2012). Ge-indistinguishability: Differential privacy for location-based systems. *arXiv preprint arXiv:1212.1984*.
- Askoy, D., y Franklin, M. (1999). Rxw: A scheduling approach to large scale on-demand broadcast. *IEEE/ACM Trans. Networking*, 7, 846–861.
- Bamba, B., Liu, L., Pesti, P., y Wang, T. (2008). Supporting anonymous location queries in mobile environments with privacygrid. En *Proceedings of the 17th international conference on world wide web* (pp. 237–246).
- Barkhuus, L., y Dey, A. K. (2003). Location-based services for mobile telephony: a study of users' privacy concerns. En *Interact* (Vol. 3, pp. 702–712).
- Chen, J., Lee, V. C., y Liu, K. (2010). On the performance of real-time multi-item request scheduling in data broadcast environments. *Journal of Systems and Software*, 83(8), 1337–1345.
- Chow, C.-Y., Mokbel, M. F., y Aref, W. G. (2009). Casper\*: Query processing for location services without compromising privacy. *ACM Transactions on Database Systems (TODS)*, 34(4), 24.

- Chow, C.-Y., Mokbel, M. F., y He, T. (2008). Tincasper: a privacy-preserving aggregate location monitoring system in wireless sensor networks. En *Proceedings of the 2008 acm sigmod international conference on management of data* (pp. 1307–1310).
- Chow, C.-Y., Mokbel, M. F., y Liu, X. (2006). A peer-to-peer spatial cloaking algorithm for anonymous location-based service. En *Proceedings of the 14th annual acm international symposium on advances in geographic information systems* (pp. 171–178).
- Dewri, R., Ray, I., Ray, I., y Whitley, D. (2008). Optimizing on-demand data broadcast scheduling in pervasive environments. En *Proceedings of the 11th international conference on extending database technology: Advances in database technology* (pp. 559–569).
- Dewri, R., Ray, I., Ray, I., y Whitley, D. (2010a). Query m-invariance: Preventing query disclosures in continuous location-based services. En *2010 eleventh international conference on mobile data management* (pp. 95–104).
- Dewri, R., Ray, I., Ray, I., y Whitley, D. (2010b). Real time stochastic scheduling in broadcast systems with decentralized data storage. *Real-Time Systems*, 45(3), 143–175.
- Fang, Q., Vrbsky, S. V., Dang, Y., y Ni, W. (2004). A pull-based broadcast algorithm that considers timing constraints. En *Workshops on mobile and wireless networking/high performance scientific, engineering computing/network design and architecture/optical networks control and management/ad hoc and sensor networks/compil* (pp. 46–53).
- Galdames, P., y Cai, Y. (2012). Efficient processing of location-cloaked queries. En *2012 proceedings ieee infocom* (pp. 2480–2488).
- Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., y Tan, K.-L. (2008). Private queries in location based services: anonymizers are not necessary. En *Proceedings of the 2008 acm sigmod international conference on management of data* (pp. 121–132).
- Ghinita, G., Kalnis, P., y Skiadopoulos, S. (2007). Prive: anonymous location-based queries in distributed mobile systems. En *Proceedings of the 16th international conference on world wide web* (pp. 371–380).
- Gruteser, M., y Grunwald, D. (2003). Anonymous usage of location-based services through spatial and temporal cloaking. En *Proceedings of the 1st international conference on mobile systems, applications and services* (pp. 31–42).
- Guo, Y., Das, S. K., y Pinotti, C. M. (2001). A new hybrid broadcast scheduling algorithm for asymmetric communication systems: push and pull data based on optimal cut-off point. En *Proceedings of the 4th acm international workshop on modeling, analysis and simulation of wireless and mobile systems* (pp. 123–130).
- Gupta, P., Muralidharan, K., Gogri, R. L., y Damodhiran, K. (2010, agosto 31). *Providing location-based services via wireless networks*. Google Patents. (US Patent 7,787,887)

- 
- Hoh, B., y Gruteser, M. (2005). Protecting location privacy through path confusion. En *First international conference on security and privacy for emerging areas in communications networks (securecomm'05)* (pp. 194–205).
- Hu, W., Fan, C., Luo, J., Peng, C., y Du, B. (2015). An on-demand data broadcasting scheduling algorithm based on dynamic index strategy. *Wireless Communications and Mobile Computing*, 15(5), 947–965.
- Hu, W., Xia, C., Du, B., y Wu, M. (2015). An on-demand data broadcasting scheduling considering the data item size. *Wireless Networks*, 21(1), 35–56.
- Hua, J., Tong, W., Xu, F., y Zhong, S. (2017). A geo-indistinguishable location perturbation mechanism for location-based services supporting frequent queries. *IEEE Transactions on Information Forensics and Security*, 13(5), 1155–1168.
- Imielinski, T., Viswanathan, S., y Badrinath, B. (1997). Data on air: Organization and access. *IEEE Transactions on knowledge and Data Engineering*, 9(3), 353–372.
- Indyk, P., y Woodruff, D. (2006). Polylogarithmic private approximations and efficient matching. En *Theory of cryptography conference* (pp. 245–264).
- Jang, M. Y., Jang, S.-J., y Chang, J.-W. (2012). A new k-nn query processing algorithm enhancing privacy protection in location-based services. En *2012 IEEE first international conference on mobile services* (pp. 17–24).
- Kalnis, P., Ghinita, G., Mouratidis, K., y Papadias, D. (2007). Preventing location-based identity inference in anonymous spatial queries. *IEEE transactions on knowledge and data engineering*, 19(12), 1719–1733.
- Kalyanasundaram, B., y Velauthapillai, M. (2003). On-demand broadcasting under deadline. En *European symposium on algorithms* (pp. 313–324).
- Khoshgozaran, A., Shirani-Mehr, H., y Shahabi, C. (2013). Blind evaluation of location based queries using space transformation to preserve location privacy. *GeoInformatica*, 17(4), 599–634.
- Kini, A. K., y Kulkarni, S. (2017). Real time implementation of k fake location generation algorithm to protect location privacy in location based services. En *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 142–148).
- Krumm, J. (2009). A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6), 391–399.
- Lee, H., Oh, B.-S., Kim, H.-i., y Chang, J. (2012). Grid-based cloaking area creation scheme supporting continuous location-based services. En *Proceedings of the 27th annual ACM symposium on applied computing* (pp. 537–543).

- Lei, M., Vrbsky, S. V., y Xiao, Y. (2010). Scheduling on-demand data broadcast in mixed-type request environments. *Computer Networks*, 54(5), 811–825.
- Li, T. C., y Zhu, W. T. (2012). Protecting user anonymity in location-based services with fragmented cloaking region. En *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)* (Vol. 3, pp. 227–231).
- Liu, F., Hamza-Lup, G., y Hua, K. A. (2010). Using broadcast to protect user privacy in location-based applications. En *2010 IEEE Globecom Workshops* (pp. 1561–1565).
- Liu, L. (2007). From data privacy to location privacy: models and algorithms. En *Proceedings of the 33rd International Conference on Very Large Data Bases* (pp. 1429–1430).
- Lu, Z., Wu, W., Li, W. W., y Pan, M. (2016). Efficient scheduling algorithms for on-demand wireless data broadcast. En *IEEE Infocom 2016—the 35th Annual IEEE International Conference on Computer Communications* (pp. 1–9).
- Lv, J., Lee, V. C., Li, M., y Chen, E. (2012). Profit-based scheduling and channel allocation for multi-item requests in real-time on-demand data broadcast systems. *Data & Knowledge Engineering*, 73, 23–42.
- Ma, C., Zhou, C., y Yang, S. (2015). A voronoi-based location privacy-preserving method for continuous query in lbs. *International Journal of Distributed Sensor Networks*, 11(3), 326953.
- Magkos, E. (2012). Cryptographic approaches for privacy preservation in location-based services: A survey. En *Systems Approach Applications for Developments in Information Technology* (pp. 273–297). IGI Global.
- Mellado, D., Fernández-Medina, E., y Piattini, M. (2007). A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards & Interfaces*, 29(2), 244–253.
- Mokbel, M. F., Chow, C.-Y., y Aref, W. G. (2006). The new casper: Query processing for location services without compromising privacy. En *Proceedings of the 32nd International Conference on Very Large Data Bases* (pp. 763–774).
- Ngo, C. N., y Dang, T. K. (2013). On efficient processing of complicated cloaked region for location privacy aware nearest-neighbor queries. En *Information and Communication Technology-Eurasia Conference* (pp. 101–110).
- Niu, B., Li, Q., Zhu, X., Cao, G., y Li, H. (2015). Enhancing privacy through caching in location-based services. En *2015 IEEE Conference on Computer Communications (Infocom)* (pp. 1017–1025).
- Pan, X., Xu, J., y Meng, X. (2011). Protecting location privacy against location-dependent attacks in mobile services. *IEEE Transactions on Knowledge and Data Engineering*,

- 24(8), 1506–1519.
- Papadias, D., Zhang, J., Mamoulis, N., y Tao, Y. (2003). Query processing in spatial network databases. *Proceedings 2003 VLDB Conference*, 802–813.
- Paulet, R., Kaosar, M. G., Yi, X., y Bertino, E. (2013). Privacy-preserving and content-protecting location based queries. *IEEE transactions on knowledge and data engineering*, 26(5), 1200–1210.
- Press, A. (2004). *Man accused of stalking ex-girlfriend with gps*. Descargado 2010-09-304, de <https://www.foxnews.com/story/man-accused-of-stalking-ex-girlfriend-with-gps>
- Riboni, D., Pareschi, L., y Bettini, C. (2009). Privacy in georeferenced context-aware services: A survey. En *Privacy in location-based applications* (pp. 151–172). Springer.
- Serjantov, A., y Danezis, G. (2002). Towards an information theoretic metric for anonymity. En *International workshop on privacy enhancing technologies* (pp. 41–53).
- Stallings, W. (2017). *Cryptography and network security: principles and practice*. Pearson Upper Saddle River.
- Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 557–570.
- Toch, E., Wang, Y., y Cranor, L. F. (2012). Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction*, 22(1-2), 203–220.
- Um, J.-H., Kim, Y.-K., Lee, H.-J., Jang, M., y Chang, J.-W. (2012). k-nearest neighbor query processing algorithm for cloaking regions towards user privacy protection in location-based services. *Journal of Systems Architecture*, 58(9), 354–371.
- Wang, J., Li, Y., Yang, D., Gao, H., Luo, G., y Li, J. (2017). Achieving effective k-anonymity for query privacy in location-based services. *IEEE Access*, 5, 24580–24592.
- Wang, J.-Y. (2012). Set-based broadcast scheduling for minimizing the worst access time of multiple data items in wireless environments. *Information Sciences*, 199, 93–108.
- Xu, J., Tang, X., y Lee, W.-C. (2006). On scheduling time-critical on-demand broadcast. *IEEE Transactions on Parallel and Distributed Systems*, 17(1), 3–14.
- Xu, T., y Cai, Y. (2009a). Feeling-based location privacy protection for location-based services. En *Proceedings of the 16th acm conference on computer and communications security* (pp. 348–357).
- Xu, T., y Cai, Y. (2009b). Location cloaking for safety protection of ad hoc networks. En *Ieee infocom 2009* (pp. 1944–1952).
- Xue, M., Kalnis, P., y Pung, H. K. (2009). Location diversity: Enhanced privacy protection in lo-

location based services. En *International symposium on location-and context-awareness* (pp. 70–87).