

MODELOS, MEDICIONES Y TARIFICACIÓN PARA REDES CON  
CALIDAD DE SERVICIO

Autor:

Pablo Belzarena

Director de Tesis:

María Simon y Gonzalo Perera

ENVIADA EN CUMPLIMIENTO PARCIAL  
DE LOS REQUERIMIENTOS PARA OBTENER EL GRADO DE  
DOCTOR EN INGENIERÍA ELÉCTRICA

de la

UNIVERSIDAD DE LA REPÚBLICA  
MONTEVIDEO, URUGUAY

2009

## **Agradecimientos**

A los tutores de la tesis María y Gonzalo.

Al IIE en su conjunto por el apoyo y el aliento. En especial a Gabriel y Víctor por librarme de diversas responsabilidades en el Departamento de Telecomunicaciones postergando sus actividades personales.

A todos con los que he trabajado durante estos años en diversos proyectos, convenios, etc. y que sin duda hicieron posible la realización de esta tesis. Muy especialmente a Andrés, Federico, Fernando, Javier, Laura, Paola y Pedro.

*para:*  
*Diego,*  
*Cecilia,*  
*Juan Ignacio,*  
*Alicia.*

# Tabla de Contenido

<b>Tabla de Contenido</b>	<b>I</b>
<b>1. Resumen</b>	<b>1</b>
1.1. Motivación y preguntas formuladas en la tesis . . . . .	1
1.1.1. Estimación de la calidad de servicio a partir de modelos de la red y del tráfico	4
1.1.2. Estimación de la calidad de servicio utilizando mediciones activas en la red .	5
1.1.3. Tarificación en redes con calidad de servicio . . . . .	6
1.2. Las contribuciones de la tesis a la solución de los problemas formulados y publicaciones asociadas . . . . .	6
<b>I Modelos y análisis de desempeño en un camino usando la teoría de grandes desvíos</b>	<b>9</b>
<b>2. Introducción</b>	<b>10</b>
<b>3. Aplicaciones de grandes desvíos al análisis de desempeño de redes</b>	<b>11</b>
3.1. Introducción . . . . .	11
3.2. Resumen de la teoría de grandes desvíos . . . . .	13
3.2.1. Introducción . . . . .	13
3.2.2. Ecuación de Lindley y teorema de Loynes . . . . .	14
3.2.3. Definiciones básicas de la teoría de grandes desvíos . . . . .	16
3.2.4. Teorema de Cramér . . . . .	17
3.2.5. Teorema de Gärtner-Ellis . . . . .	18
3.2.6. Teorema de Mogulskii . . . . .	20
3.2.7. Principio de Contracción . . . . .	21
3.3. El régimen asintótico de buffer grande . . . . .	22
3.3.1. El análisis de un enlace . . . . .	22
3.3.2. Extensión de los resultados anteriores al análisis de una red . . . . .	24
3.4. El régimen asintótico de muchas fuentes . . . . .	26
3.4.1. Ancho de Banda Efectivo . . . . .	26
3.4.2. Probabilidad de pérdida en un enlace . . . . .	27

3.4.3.	De un enlace al análisis de una red, la propuesta de Wischik . . . . .	30
3.5.	La asintótica de muchas fuentes y buffer pequeño . . . . .	31
3.5.1.	Análisis de desempeño de una red alimentada por muchos flujos y buffer pequeño . . . . .	31
3.6.	Conclusiones . . . . .	35
<b>4.</b>	<b>Análisis mediante el método de la red ficticia</b>	<b>36</b>
4.1.	El análisis mediante la red ficticia . . . . .	36
4.2.	Condiciones y cotas del error usando el análisis de la red ficticia . . . . .	37
4.2.1.	Introducción . . . . .	37
4.2.2.	Condición suficiente . . . . .	38
4.2.3.	Condición suficiente en función del ancho de banda disponible . . . . .	41
4.2.4.	Condición suficiente pero no necesaria . . . . .	41
4.2.5.	Cota para el error . . . . .	42
4.2.6.	La cota del error en un caso particular . . . . .	43
4.3.	Ejemplos numéricos . . . . .	45
4.4.	Conclusiones . . . . .	52
<b>5.</b>	<b>Estimación del desempeño y diseño de un enlace</b>	<b>54</b>
5.1.	Estimación del ancho de banda efectivo . . . . .	54
5.1.1.	Estimación del ancho de banda efectivo . . . . .	55
5.1.2.	Estimadores no paramétricos . . . . .	55
5.1.3.	Estimadores paramétricos . . . . .	55
5.1.4.	Simulación de los estimadores paramétricos y no paramétricos . . . . .	56
5.1.5.	Dificultades para la aplicación del modelo paramétrico markoviano a trazas reales . . . . .	59
5.2.	Estimación del punto de operación de un enlace . . . . .	63
5.3.	Estimación de los parámetros de calidad de servicio de un enlace . . . . .	67
5.4.	Diseño de un enlace basado en la estimación del ancho de banda efectivo . . . . .	68
5.5.	Conclusiones . . . . .	71
<b>II</b>	<b>Estimación del desempeño en un camino usando mediciones activas</b>	<b>73</b>
<b>6.</b>	<b>La estimación de parámetros de extremo a extremo en Internet</b>	<b>74</b>
6.1.	Introducción . . . . .	74
6.2.	Revisión de algunas propuestas en el área de mediciones activas . . . . .	76
<b>7.</b>	<b>Mediciones de extremo a extremo y su aplicación a enlaces celulares</b>	<b>77</b>
7.1.	Introducción . . . . .	77
7.2.	Estimación de parámetros en enlaces celulares GPRS/EDGE . . . . .	77
7.2.1.	Introducción . . . . .	77
7.2.2.	Características de las mediciones de desempeño sobre enlaces celulares . . . . .	78
7.2.3.	Metodología de medición . . . . .	80

7.2.4.	Estimación de la capacidad de un enlace celular . . . . .	81
7.2.5.	Resultados . . . . .	84
7.2.6.	Round Trip Time . . . . .	87
7.3.	Throughput vs Capacidad: influencia de diversas características de un enlace celular	88
7.3.1.	Estudio throughput/capacidad en un enlace GPRS/EDGE . . . . .	89
7.3.2.	Throughput y consideraciones sobre el desempeño de una red 3G/HSDPA. . . . .	92
7.3.3.	Pruebas UTRAN. . . . .	100
7.4.	Conclusiones . . . . .	104
<b>8.</b>	<b>Aprendizaje estadístico y su aplicación al monitoreo y control de admisión en redes</b>	<b>108</b>
8.1.	Introducción . . . . .	108
8.2.	Aprendizaje estadístico . . . . .	110
8.2.1.	Introducción a los modelos de regresión . . . . .	110
8.2.2.	El estimador de Nadaraya-Watson para variables en $\mathbb{R}^d$ . . . . .	110
8.2.3.	EL estimador de Nadaraya-Watson para variables funcionales . . . . .	112
8.3.	Extensiones del estimador funcional de Nadaraya-Watson al caso no estacionario . . . . .	113
8.4.	Estimación de la QoS de extremo a extremo de una aplicación . . . . .	116
8.4.1.	Análisis del problema y metodología de estimación . . . . .	116
8.4.2.	Estimación de la QoS a partir de la distribución de tiempos entre paquetes de prueba . . . . .	120
8.5.	Conclusiones . . . . .	124
8.6.	Apéndice: Demostraciones . . . . .	126
8.6.1.	Prueba del lema 8.1 . . . . .	126
8.6.2.	Prueba del Teorema 8.2 . . . . .	127
<b>9.</b>	<b>Estimación de la QoS utilizando un estimador de la cola del enlace y Support Vector Machines</b>	<b>130</b>
9.1.	Introducción . . . . .	130
9.2.	Support Vector Machines . . . . .	130
9.2.1.	SVM para clasificación . . . . .	130
9.2.2.	SVM en regresión . . . . .	134
9.3.	Estimador de la cola del enlace y del estado de un camino en la red . . . . .	135
9.3.1.	El caso de un enlace . . . . .	135
9.3.2.	Extensión a un camino en la red . . . . .	138
9.4.	Regresión utilizando el estimador de la cola o del estado del camino . . . . .	139
9.5.	Regresión con SVM utilizando parámetros de la estimación de la cola o del camino . . . . .	142
9.6.	Estimación con diferentes videos . . . . .	145
9.7.	Conclusiones . . . . .	147
<b>10.</b>	<b>Software de medición y estimación de la QoS</b>	<b>150</b>
10.1.	Introducción . . . . .	150
10.2.	Metronet . . . . .	150

10.2.1. Especificación del software: restricciones y requerimientos. . . . .	151
10.2.2. Uso del software. . . . .	152
10.2.3. Arquitectura y diseño de MetroNet. . . . .	154
10.3. MetroRed, una red overlay de mediciones distribuida . . . . .	155
10.3.1. Paquetes que componen MetroRed . . . . .	158
10.4. Conclusiones. . . . .	166
<b>11. Resultados experimentales. . . . .</b>	<b>167</b>
11.1. Introducción . . . . .	167
11.2. Pruebas ADSL . . . . .	167
11.3. Prueba a través de múltiples dominios . . . . .	169
11.4. Prueba a través de un enlace celular . . . . .	173
11.5. Conclusiones . . . . .	176
<b>12. Calidad de servicio percibida . . . . .</b>	<b>178</b>
12.1. Introducción . . . . .	178
12.2. Métodos para evaluar la PQoS . . . . .	180
12.2.1. Evaluación subjetiva . . . . .	180
12.2.2. Evaluación objetiva - Métodos intrusivos . . . . .	181
12.2.3. Evaluación Objetiva - Métodos no-intrusivos . . . . .	183
12.3. Evaluación experimental y resultados . . . . .	183
12.3.1. La maqueta de pruebas utilizada . . . . .	183
12.3.2. Resultados de los test subjetivos . . . . .	185
12.3.3. Evaluación de las diferentes técnicas . . . . .	185
12.4. Consideraciones sobre la PQoS y su integración con la estimación usando paquetes de prueba . . . . .	188
12.5. Clasificación y selección de características en regiones de QoS . . . . .	190
12.5.1. Clasificación en zonas de QoS . . . . .	190
12.5.2. Selección de variables utilizando F-Score . . . . .	191
12.6. Conclusiones . . . . .	194
<b>III Tarificación en redes con calidad de servicio . . . . .</b>	<b>201</b>
<b>13. Tarificación en redes con calidad de servicio asegurada . . . . .</b>	<b>202</b>
13.1. Introducción . . . . .	202
13.2. Redes overlay . . . . .	205
13.2.1. Service overlay networks . . . . .	206
13.3. Redes triple play . . . . .	207
13.4. Introducción a los mecanismos de subastas . . . . .	208
13.5. Planteo del problema y análisis de subastas en un enlace . . . . .	210
13.5.1. Asignación óptima como un Markov Decision Process . . . . .	212
13.6. Aproximación mediante la reducción del horizonte . . . . .	222
13.7. Conclusiones . . . . .	227

<b>14. Asignación óptima de recursos en una red</b>	<b>229</b>
14.1. Introducción . . . . .	229
14.2. El problema de optimización instantánea de los recursos de una red . . . . .	229
14.2.1. Notación y formulación del problema . . . . .	229
14.2.2. Relajación convexa y solución distribuida . . . . .	231
14.3. Una aproximación fluida al problema de horizonte reducido . . . . .	234
14.4. Subastas periódicas en el caso de una red . . . . .	236
14.5. Implementación y simulaciones . . . . .	238
14.5.1. Escenario 1: subastas de la capacidad de un enlace. . . . .	240
14.5.2. Escenario 2: Red lineal . . . . .	241
14.5.3. Escenario 3: Red overlay. . . . .	242
14.6. Consideraciones sobre la estrategia de los usuarios y la teoría de juegos . . . . .	244
14.7. Conclusiones . . . . .	245
<b>IV Conclusiones de la tesis</b>	<b>247</b>
<b>15. Conclusiones</b>	<b>248</b>
<b>Bibliografía</b>	<b>250</b>



# Capítulo 1

## Resumen

### 1.1. Motivación y preguntas formuladas en la tesis

Servicios exigentes en calidad de servicio (QoS) como video bajo demanda, video conferencias de alta calidad, servicios telemáticos con requerimientos de tiempo real, etc. se han desplegado a un ritmo menor que el inicialmente esperado. Entre las causas de esta situación se encuentra la dificultad que existe para garantizar calidad de servicio (QoS) en redes ip. También, que los operadores por diferentes razones, no han adoptado las soluciones propuestas para brindar QoS planteadas en los últimos 15 años (IntServ, DiffServ, Flow Aware, etc.). Estas dificultades se han acrecentado además por la creciente heterogeneidad de las redes de acceso sobre las que se deben ofrecer estos servicios (xdsl, cablemodem, wifi, wimax, 2G, 3G, redes mesh, etc.). Para el tipo de servicios referido, es importante señalar que lo que interesa es analizar la calidad de servicio de extremo a extremo en la red, es decir, desde un usuario hasta otro usuario o hasta el equipo que le brinda el servicio. Aún cuando se tuviera acceso a todos los nodos de un camino entre dos extremos, no es fácil concluir del desempeño individual de cada nodo el desempeño de extremo a extremo del camino. El problema se vuelve menos manejable aún cuando se quiere ofrecer estos servicios a través de múltiples dominios, donde los nodos del camino se encuentran además bajo la administración de diferentes operadores de red.

La debilidad para ofrecer servicios con QoS garantizada tiene origen en la arquitectura de Internet creada con el objetivo de ofrecer servicios básicos como el mail y el intercambio de archivos. En los últimos años, como se mencionó, se ha desarrollado múltiples propuestas para brindar QoS y aún se siguen estudiando nuevos modelos de Internet o “la arquitectura de la Nueva Internet”. De este conjunto de propuestas los operadores han adoptado muy pocos cambios. Los principios básicos de la vieja arquitectura se han mantenido. Lo que han hecho los operadores para brindar servicios con algún tipo de requerimiento de QoS ha sido sobredimensionar el backbone, hacer crecer el ancho de banda del acceso a la red e implementar algunas modificaciones en la arquitectura de la red de acceso para separar los servicios con requerimientos de QoS. Dentro de los cambios implementados en la arquitectura de la red de acceso para los operadores que brindan servicios “premium”, se encuentra la utilización de mecanismos de control de admisión . Esto se observa por ejemplo en los servicios de video streaming sobre redes celulares, en servicios de video llamada o en

la configuración de límites en la cantidad simultánea de conexiones HSPA a los efectos de poder asegurar un mínimo de ancho de banda en el acceso a cada usuario 3G. También se implementan mecanismos de control de admisión en múltiples propuestas de arquitectura comercial para redes “triple play” (por ejemplo las propuestas de CISCO y ALCATEL incluyen este tipo de mecanismos) o en servicios de video bajo demanda.

Estas arquitecturas se basan en contar con un backbone sobredimensionado y por tanto, los mecanismos de control de admisión se basan sólo en la existencia o no de recursos en la red de acceso (recursos de radio frecuencia y canales asociados en el caso celular, cantidad de conexiones en el enlace de acceso en el caso de redes triple play, etc.). En la medida que los mecanismos de control de admisión se limitan a la existencia de recursos en un “enlace”, políticas relativamente simples pueden ser implementadas para este fin.

Un problema más interesante se plantea cuando el mecanismo de control de admisión debe basarse en la QoS que se le puede asegurar al nuevo usuario de extremo a extremo. Esta calidad no depende sólo del estado del enlace de acceso, sino de todo el camino hasta el otro extremo donde se brinda el servicio. Si bien es cierto que al estar el backbone sobredimensionado los problemas debieran darse principalmente en el acceso, esto no es cierto en todas las circunstancias y para todos los parámetros que impactan en la QoS en redes tan complejas como Internet. Por otro lado, en la medida que el ancho de banda del acceso continúa creciendo y la cantidad de usuarios aumentando, la hipótesis de backbone sobredimensionado no es evidente que pueda ser sostenida en el futuro. Además, cuando los servicios se desean ofrecer a través de múltiples dominios, pueden aparecer otros puntos donde se puede deteriorar la QoS y que no son sólo en el acceso. Las políticas de interconexión de dominios, en general no se basan en criterios de QoS sino en criterios de costos de los operadores y en algunos casos se detecta en los puntos de interconexión problemas con la QoS.

El problema de implementar mecanismos de control de admisión ha sido desarrollado extensamente en la literatura de redes. En particular el área de control de admisión basada en mediciones ha centrado la atención de diversos autores durante los últimos años. Los primeros trabajos en control de admisión basados en mediciones se centran en lo que se puede llamar control de admisión individual por enrutador. Es decir que cada enrutador mide alguna característica del tráfico que recibe (en general son magnitudes de peor caso como el pico del tráfico) y con esta medida y un umbral de admisión decide si se acepta o no un nuevo flujo. Una reseña de diferentes trabajos en este tipo de mecanismos y una comparación de los mismos se encuentra en el trabajo [26]. Como se mencionó antes, un problema más interesante es lograr que las decisiones de admisión se tomen evaluando el desempeño del camino a utilizar por el nuevo flujo y no individualmente en cada enrutador. En los últimos años se han propuesto varios mecanismos de control de admisión basados en mediciones que evalúan el desempeño de extremo a extremo. Algunas propuestas se basan en estimar el desempeño a partir de información agregada a los paquetes en los enrutadores intermedios como por ejemplo el marcado del bit ECN (Explicit Congestion Notification) (ver por ejemplo [81]). Estas propuestas requieren la intervención de todos los enrutadores intermedios para que las decisiones en los extremos sean acertadas, lo cual no es totalmente deseable. Otras propuestas se basan en enviar tráfico, y medir pérdidas o jitter sobre este tráfico para decidir si se acepta o no un nuevo flujo [23, 77, 101]. Estas propuestas en algunos casos envían paquetes de prueba y de acuerdo a las pérdidas o el jitter de dichos paquetes deciden si dejan ingresar un nuevo flujo de la aplicación. Esto tiene el problema de que el desempeño que experimenta una aplicación y el tráfico

de prueba no tiene porqué ser el mismo. Para solucionar este inconveniente, en otros trabajos se propone enviar el tráfico de la aplicación durante un período y estimar el desempeño sobre el propio tráfico de la aplicación. En algunos casos, el tráfico de la aplicación puede deteriorar el desempeño de los flujos previamente aceptados y que están utilizando la red. Por este motivo, por ejemplo para el acceso de ciertas redes con ancho de banda limitado esta técnica no es razonable. En la tesis se analizará cómo estimar el desempeño de una aplicación de extremo a extremo pero proponiendo metodologías que mejoren en algunos aspectos los mecanismos anteriores.

Otro factor importante para analizar este problema, es que a los operadores se les ha hecho difícil encontrar nuevos modelos de negocio que justifiquen los costos e inversiones que implican las inversiones requeridas en la red. El modelo de cobro por tarifa plana, ampliamente difundido, no contribuye a mejorar la situación de la calidad de servicio en la red. Este mecanismo no brinda incentivos al usuario para cuidar el uso de los recursos. Si se desea que el usuario realice un uso adecuado de los recursos compartidos en la red, es importante buscar mecanismos económicos que incentiven este tipo de conductas.

En el contexto reseñado, el problema general que se analiza en la tesis es cómo obtener información de la calidad de servicio a lo largo de un camino a los efectos de implementar mecanismos de control de admisión basados no sólo en la información de un enlace sino en la de todo el camino. Las principales preguntas que se analizan en la tesis son entonces las siguientes: ¿Es posible predecir la calidad de servicio de extremo a extremo que recibirá un nuevo flujo? ¿Cómo puede complementarse el control de admisión con la optimización de la ganancia de este operador que ofrece servicios “premium” con calidad de servicio?

Para analizar estas preguntas se dividió el trabajo en tres partes y la tesis está organizada con ese criterio. En cada parte se analizan las siguientes preguntas:

- Parte 1: ¿Qué modelo se podría utilizar para analizar la calidad de servicio en un camino de una red? ¿Qué limitaciones tiene este modelo? ¿Puede utilizarse ese modelo para estimar la QoS de extremo a extremo en línea?
- Parte 2: ¿Cómo estimar la calidad de servicio de utilizando mediciones activas en la red? ¿Es posible predecir la calidad de servicio de un servicio como el video sin cargar la red con las medidas?
- Parte 3: ¿Cómo optimizar la ganancia del operador en una red que ofrece servicios premium con calidad de servicio?

En lo que sigue se explica el problema estudiado en cada parte de la tesis. Previamente a analizar cada parte, es importante señalar lo siguiente. Una de las motivaciones de la tesis es el análisis de la calidad de servicio a lo largo de un camino a los efectos de utilizar esa información en mecanismos de control de admisión. Sin embargo, muchas de las técnicas desarrolladas en la tesis tienen un área de aplicación más general que el problema particular de control de admisión. Por ejemplo, otras posibles aplicaciones pueden ser monitorear la calidad de servicio en un camino de la red para implementar algoritmos de control de las aplicaciones o verificar un acuerdo de nivel de servicio (SLA), u obtener información para analizar las tendencias o variaciones de la calidad de servicio en a lo largo del tiempo, etc.

### 1.1.1. Estimación de la calidad de servicio a partir de modelos de la red y del tráfico

Una de las razones por lo que ha sido difícil poner en práctica diversas propuestas de arquitectura para asegurar la QoS en Internet es la falta de modelos matemáticos de la red y del tráfico que permitan abordar situaciones complejas y realistas. En el área de modelado en general los resultados más importantes están limitados a estudiar un enlace y haciendo simplificaciones para los modelos de tráfico. Cuando se desea estudiar el desempeño de extremo a extremo de un camino en la red y no sólo de un enlace, en la literatura se encuentran pocos resultados aplicables a Internet. Los primeros resultados en esta área fueron las llamadas redes de Jackson o modelos derivados de éstas. En estos casos las hipótesis sobre las distribuciones de los tiempos de arribo y servicio en general limitan la aplicabilidad de estos modelos a Internet. Un área que ha tenido un importante desarrollo en los últimos 15 años ha sido el de las aplicaciones de la teoría de grandes desvíos a las redes de datos. La teoría de grandes desvíos tiene interés porque permite utilizar modelos bastante generales para el tráfico y además, porque muchos de los parámetros de calidad de servicio que interesan corresponden a valores asociados a las colas de las distribuciones (la probabilidad de que un buffer se llene por encima de cierto umbral, la probabilidad de que el retardo sea mayor que cierto valor, etc.). La teoría de grandes desvíos se centra en analizar el comportamiento de este tipo de fenómenos asociados a las colas de las distribuciones. En la aplicación de la teoría de grandes desviaciones al análisis de redes de Telecomunicaciones se ha trabajado principalmente sobre tres asintóticas. La asintótica de “buffer grande”, donde se analiza el comportamiento límite de la probabilidad de desborde cuando el buffer del enlace tiende a infinito. La asintótica de “muchas fuentes”, donde se estudia el comportamiento límite cuando se hace tender a infinito el número de usuarios conjuntamente con la capacidad y el buffer del enlace. La asintótica de “muchas fuentes y buffer pequeño”, donde se hace tender a infinito el número de usuarios junto con la capacidad del enlace, pero el buffer del enlace crece más lentamente que el número de usuarios.

Aún en este contexto, existen pocas propuestas que permitan analizar un camino con varios enlaces en una red. La mayoría de los resultados se aplican a un enlace o a topologías particulares de red. Ozurk, Mazumdar y Likhanov plantean un modelo utilizando la asintótica de muchas fuentes y buffer pequeño que permite obtener resultados de desempeño para un camino en la red. Sin embargo, para ser aplicado en línea (por ejemplo para control de admisión) requiere de algoritmos recursivos que pueden tener un importante costo computacional en una red compleja. En el marco de la teoría de grandes desvíos, en esta tesis se presentan algunos resultados que permiten diseñar un enlace basado en los requerimientos de calidad de servicio y además se estudia cuando es posible simplificar el método de análisis propuesto por Ozturk et al. y obtener algoritmos más simples para calcular algunos parámetros de calidad de servicio.

Aún con estos modelos, para poder aplicarlos en línea, es necesario estimar la función de velocidad de gran desvío a partir de medidas de tráfico. Este problema es muy amplio y en varios aspectos ha sido escasamente analizado en la literatura. En esta parte de la tesis se analizan también diferentes aspectos de este problema de estimación.

Sin embargo, si bien mediante el modelado con la teoría de grandes desvíos es posible analizar la calidad de servicio en un camino de la red, estos resultados se aplican básicamente al estudio de un camino en el backbone. Por lo tanto, en un camino que involucre el pasaje a través de la red de

acceso no es posible aplicar estos modelos. Si bien hay resultados de la teoría de grandes desvíos que se pueden aplicar a un enlace del acceso, no es posible integrar el análisis del backbone y el de la red de acceso mediante la teoría de grandes desvíos (este punto se verá con más detalle en el capítulo 3). Por este motivo, para casos que no involucren solamente enrutadores del backbone, el modelado anterior no es suficiente. Una alternativa es estimar el desempeño utilizando mediciones activas en la red (es decir : enviar paquetes de prueba y analizar como afecta a dichos paquetes el pasaje a través de la red). Este problema se analiza en la siguiente parte de la tesis.

### **1.1.2. Estimación de la calidad de servicio utilizando mediciones activas en la red**

En esta parte de la tesis, el objetivo es estimar por ejemplo parámetros de desempeño como el retardo o el jitter que experimenta una aplicación (por ej. un video bajo demanda) al ser transmitido por un camino de la red. Un técnica posible para tal fin es enviar el video, analizar sus paquetes y calcular el parámetro de interés (jitter por ejemplo). Sin embargo, esta propuesta no es adecuada en muchos contextos como en el de este trabajo. Si el objetivo es decidir si se admite o no el flujo y para hacerlo se envía el video para medir el jitter, se puede deteriorar la calidad de servicio al agregar este nuevo flujo, y se estará perjudicando al conjunto de los usuarios que están usando este camino.

Para evitar este problema, muchos sistemas de monitoreo lo que hacen es mandar una serie de paquetes de pruebas “livianos” que cargan poco la red. Se mide el jitter por ejemplo sobre los paquetes de prueba y se traslada el resultado obtenido a lo que se espera que le suceda al video. El problema es precisamente que los paquetes de prueba no cargan la red y por lo tanto no es necesariamente cierto, por ejemplo, que si ellos experimentan poco jitter el video también experimentará poco jitter.

En la tesis se propone una técnica que se posiciona en un punto intermedio entre ambas formas de estimación. La metodología propuesta se basa en utilizar herramientas de aprendizaje estadístico. La idea es enviar paquetes de prueba que carguen poco la red y analizar la estadística de las variaciones de tiempo de estos paquetes. Con esta información estadística se caracteriza el estado del camino de la red y se aprende la relación entre dicho estado y el parámetro de desempeño de interés sobre el video cuando la red está en ese estado. Luego de esta etapa de aprendizaje, para monitorear la red y decidir si se acepta o no un nuevo flujo, sólo es necesario monitorear el estado del camino enviando los paquetes de prueba livianos y (sin enviar el video) estimar cuál sería el desempeño que se obtendría si se enviara el video. Además de estos problemas generales a cualquier camino en la red, cuando el enlace de acceso es celular aparecen problemas adicionales. Algunos problemas que afectan la calidad de servicio son por ejemplo la capacidad variable con el tiempo y el desempeño de diversos protocolos sobre este tipo de enlaces. Por este motivo, en esta parte de la tesis, se estudian las particularidades de este tipo de enlaces y se proponen métodos de medición y estimación.

Las preguntas de esta parte son entonces ¿es posible aprender la relación entre el desempeño de los paquetes de prueba y la QoS de extremo a extremo que experimenta una aplicación? ¿es posible hacerlo en una red como Internet donde el tráfico es no estacionario ? ¿qué consideraciones adicionales se agregan en el caso de acceso celular? ¿cuáles son los límites de estas técnicas de

estimación y medición?

### 1.1.3. Tarificación en redes con calidad de servicio

Por último en esta tercera parte se enfoca el problema de redes con calidad de servicio asegurada pero considerando además el aspecto económico de las mismas. El problema ahora se formula de la siguiente manera. El operador al igual que antes, debe decidir a cuántos usuarios puede dejar ingresar a la red de calidad asegurada (control de admisión). Pero además, en esta parte se agrega el objetivo de dejar ingresar a aquel conjunto de usuarios que maximicen la ganancia del operador. Para entender este problema, se puede primero analizar el caso simple de un solo enlace, y donde lo que se desea es optimizar lo que gana el operador en el momento que admite a el o los usuarios que quieren usar el servicio. La decisión en este caso será simplemente:

- aceptar la máxima cantidad de usuarios compatibles con las restricciones de calidad de servicio
- si hay más usuarios que los que se puede admitir para garantizar la calidad de servicio, dejar ingresar a aquellos que estén dispuestos a pagar más por usar la red (por ver el video por ejemplo).

Si este caso se extiende a una red con una topología general, se transforma en un problema más complejo porque los usuarios que usan diferentes rutas comparten los recursos de la red. En este tipo de problemas además, para que pueda ser escalable, se busca que la solución sea totalmente distribuida. El no tener un controlador central que tome las decisiones, agrega más complejidad al problema.

Por otro lado, lo que busca un operador como objetivo es maximizar la ganancia media a largo plazo y no maximizar la ganancia actual a riesgo de perder ganancia futura. Es decir, al operador puede convenirle en cierto momento no ocupar toda la capacidad de la red y dejar capacidad disponible para futuros usuarios que puedan querer usarla. Este tipo de decisión puede ser importante si lo que están dispuestos a pagar los usuarios actuales no es conveniente comparado con lo que el operador espera que estén dispuestos a pagar los usuarios futuros. En este contexto, las preguntas para esta parte final del trabajo son las siguientes. Si se agrega al problema de control de acceso, el problema de maximizar la ganancia del operador ¿es posible encontrar soluciones distribuidas a este problema en una red con topología arbitraria? ¿qué dificultades presenta? ¿es posible para el caso de un enlace maximizar la ganancia media futura del operador? ¿es posible maximizar la ganancia media futura de forma distribuida en el caso de una red?

## 1.2. Las contribuciones de la tesis a la solución de los problemas formulados y publicaciones asociadas

En esta sección se resumen las principales contribuciones realizadas a lo largo de la tesis. Las mismas se separarán la tres áreas de trabajo descritas anteriormente.

En el área de modelado, se trabajó sobre un modelo simplificado de red llamado “red ficticia” que permite un cálculo mucho más simple y que asegura la calidad de servicio sobre la red real. Se

encontró una condición suficiente para que el análisis de desempeño calculado sobre la red ficticia y sobre la real coincidan. Además, se encontraron cotas para el error en el caso de no coincidencia. Estos resultados fueron publicados en la conferencia:

*Pablo Belzarena; Paola Bermolen; Pedro Casas; María Simon, Virtual Path Networks Fast Performance Analysis, Heterogeneous Networks Conference HetNet'04, Ickley, UK.*

y en su versión extendida como capítulo del libro:

*Pablo Belzarena; Paola Bermolen; Pedro Casas; María Simon, Virtual Path Networks Fast Performance Analysis, título del libro: Mobility Management and Quality-of-Service for Heterogeneous Networks, Capítulo 17, River Publishers, 2009.*

En esta área se trabajó también en el problema de estimar la función de gran desvío y los parámetros de desempeño en el régimen de muchas fuentes o muchas fuentes y buffer pequeño. A partir de estas estimaciones, se plantea una metodología para diseñar un enlace de la red (dimensionar capacidad y buffer) y definir un intervalo de confianza para dicho dimensionado. Estos resultados han sido publicados en:

*Laura Aspirot; Pablo Belzarena; Paola Bermolen; Gonzalo Perera; María Simon, Quality of Service parameters and link operation point estimation based on effective Bandwidth- extended version. Elsevier: Journal on Performance Evaluation. Special Issue on Performance Modeling and Evaluation of Heterogeneous Networks Volume 59, Issues 2-3, February 2005.*

Se debe mencionar también que esta parte del trabajo se desarrolló en el marco de un proyecto de investigación concursable Fondo Clemente Estable “Ingeniería de tráfico y Calidad de Servicio en redes IP/MPLS”) del que fui co-responsable. También se contó con el apoyo del proyecto ECOS-SUD: “Internet como red convergente, ingeniería de tráfico, mediciones y tarifación”, cuyo responsable por la Facultad de Ingeniería fue Hector Cancela.

En el área de mediciones para estimar la calidad de servicio de extremo a extremo, se trabajó en el marco de dos proyectos de investigación concursables del Programa de Desarrollo Tecnológico, de los cuales he sido responsable: proyecto MetroNet PDT 17/02 y MetroNet II PDT 46/03. Además, los resultados de estos trabajos de investigación se aplicaron en dos convenios entre ANTEL y la Facultad de Ingeniería, referidos a la red GPRS/EDGE y a la red 3G/HSDPA de ANTEL.

Los resultados en esta área tienen diferente naturaleza. Por un lado se desarrolló una metodología de medición y estimación basada en técnicas de aprendizaje estadístico para estimar la calidad de servicio de extremo a extremo utilizando paquetes de prueba. Esta técnica se basa en métodos de regresión no paramétrica. Para esto se extendieron resultados de convergencia del estimador de Nadaraya-Watson funcional para el caso no estacionario.

Los resultados de esta parte del trabajo fueron publicados en la conferencia:

*Pablo Belzarena; Laura Aspirot; Gonzalo Perera; Bruno Bazzano, End-To-End Quality of Service Prediction Based On Functional Regression, 2005, Heterogeneous Networks Conference, HET-NET 05, Ickley, UK .*

y en su versión extendida como capítulo del libro:

*Pablo Belzarena; Laura Aspirot; Gonzalo Perera; Bruno Bazzano, End-To-End Quality of Service Prediction Based On Functional Regression, título del libro: Mobility Management and Quality-of-Service for Heterogeneous Networks, Capítulo 19, River Publishers, 2009.*

Posteriormente se modificaron estas técnicas, buscando otros métodos para estimar el estado de la red y otras técnicas de aprendizaje estadístico como Support Vector Machines. Estos últimos

resultados aún no han sido publicados.

También se aplicaron técnicas de medición y estimación para estimar la capacidad de un enlace celular. Estos resultados fueron publicados en:

*Pablo Belzarena; Javier Pereira; Juan Negreira; Santiago Pérez, End-to-End Measurements over GPRS-EDGE Networks. In: IFIP/ACM Latin America Networking Conference, LANC 2007, 2007, San José, Costa Rica. Memorias de la XXXIII Conferencia Latinoamericana en Informática. 2007.*

También se desarrollaron herramientas de medición sobre Internet que implementaron las técnicas de aprendizaje estadístico mencionadas. Una primera versión de este software fue publicada en el trabajo:

*Pablo Belzarena; Víctor González-Barbone; Federico Larroca; Pedro Casas., Metronet: software para medición de calidad de servicio en voz y video. 2006, Congreso Iberoamericano de Telemática, CITA 2006, México.*

Una versión posterior del software que plantea una arquitectura distribuida de nodos overlay de mediciones, se denomina MetroRed. Con MetroRed se realizaron mediciones que se presentan como resultados en algunos capítulos de la tesis. En este momento se está preparando un artículo para publicar estos últimos resultados.

Si bien el foco de este trabajo no se centra en estimar la calidad de servicio percibida por el usuario (PQoS), varias técnicas en esta área tienen puntos de contacto con el trabajo desarrollado en la tesis y pueden complementarse. Por lo tanto, se analizará la relación entre las técnicas de estimación de PQoS con los métodos de estimación desarrolladas en la tesis. En la siguiente publicación se analizan y comparan diferentes técnicas de análisis de calidad de servicio percibida:

*Pedro Casas; Pablo Belzarena; Sandrine Vaton, End-2-End Evaluation of IP Multimedia Services, a User Perceived Quality of Service Approach. In: 18th ITC Specialist Seminar on Quality of Experience, 2008, Karlskrona, Sweden. Proceedings of the 18th ITC Specialist Seminar on Quality of Experience.*

En el área de tarificación en redes con calidad de servicio, esta tesis se desarrolló en el marco del proyecto de investigación del Programa de Desarrollo Tecnológico, Tarificación en redes multiservicio, PDT 69/03. Las contribuciones de esta tesis en esta área son las siguientes. Formular el problema de optimizar la ganancia media futura del operador en una red con calidad de servicio, utilizando subastas como mecanismo de tarificación. Formular el problema anterior como un Markov Decision Process. Encontrar una aproximación para resolver este problema de forma totalmente distribuida en una red con topología general. Estos resultados fueron publicados en los siguientes artículos:

*Pablo Belzarena; Andres Ferragut; Fernando Paganini, Auctions for resource allocation in overlay networks. In: Network Control and Optimization, Lecture Notes in Computer Science, aceptado para ser publicado en 2009.*

*Pablo Belzarena; Andres Ferragut; Fernando Paganini, Network bandwidth allocation via distributed auctions with time reservations, IEEE INFOCOM 2009, Rio de Janeiro, Abril 2009.*



## Parte I

# Modelos y análisis de desempeño en un camino usando la teoría de grandes desvíos

## Capítulo 2

# Introducción

En esta parte de la tesis se analizará la posibilidad de utilizar la teoría de grandes desvíos para realizar control de admisión en línea teniendo en cuenta el desempeño de extremo a extremo en la red. Para tal fin, es necesario contar con un método que permita fácilmente y en línea decidir si es posible o no aceptar un flujo. El método de la red ficticia analizado en esta parte de la tesis puede ser usa en ese contexto. Además, para poder hacer control de admisión en línea es necesario estimar los modelos de tráfico y las funciones de gran desvío. Este último punto será otro de los temas analizados en esta parte de la tesis.

Se comenzará en el capítulo 3 reseñando varios resultados importantes de diversos autores y se analizarán las posibilidades y limitaciones del análisis de desempeño utilizando la teoría de grandes desvíos.

En los capítulos 4 y 5 se presentarán los resultados de esta tesis relacionados con la aplicación de grandes desvíos.

En el capítulo 4 se analiza cuándo es posible simplificar el análisis de desempeño de una red mediante el método de la red ficticia que se introducirá en este capítulo. Este método simplifica el análisis de desempeño y habilita a que sea utilizado en línea por ejemplo para realizar control de admisión. En dicho capítulo se encuentran condiciones bajo las cuales se asegura que los resultados del análisis de la red ficticia son exactos y también se encuentran cotas para el error cuando dichas condiciones no se verifican.

En el capítulo 5 se analiza la estimación a partir de mediciones de los parámetros de desempeño de un enlace o una red cuando se utilizan técnicas provenientes de la teoría de grandes desvíos. La mayoría de los trabajos de grandes desvíos aplicados a redes se enfocan en calcular la función de velocidad de gran desvío para los parámetros de desempeño de una red. Sin embargo, no hay muchos trabajos que enfoquen problemas de estimación de estos parámetros a partir de mediciones. El objetivo del capítulo 5 será analizar algunas técnicas de estimación en grandes desvíos, las dificultades que se encuentran en esta área y proponer métodos para estimar el desempeño y diseñar una red, utilizando resultados que se expondrán en dicho capítulo.

## Capítulo 3

# Aplicaciones de grandes desvíos al análisis de desempeño de redes

### 3.1. Introducción

En la sección 3.2 de este capítulo, se presenta la notación, definiciones y un resumen de los principales resultados de la teoría de grandes desvíos que se aplican en las secciones siguientes.

En la aplicación de la teoría de grandes desvíos al análisis de redes de telecomunicaciones se ha trabajado principalmente sobre tres asintóticas: la asintótica de “buffer grande” y las asintóticas de “muchas fuentes” y la de “muchas fuentes y buffer pequeño”.

La primera asintótica se estudia en la sección 3.3. En esta asintótica se analiza el desempeño de un enlace (probabilidad de pérdida de paquetes o distribución del retardo de los paquetes, por ejemplo) cuando el tamaño del buffer tiende a infinito (figura 3.1).

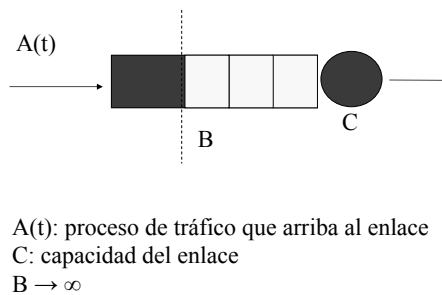


Figura 3.1: Régimen asintótico de buffer grande

La asintótica de muchas fuente se estudia en la sección 3.4. En este caso se analiza el desempeño de un enlace cuando se encuentra alimentado por un número de fuentes que tiende a infinito, escalando el buffer y la capacidad del enlace con el crecimiento de las fuentes (figura 3.2).

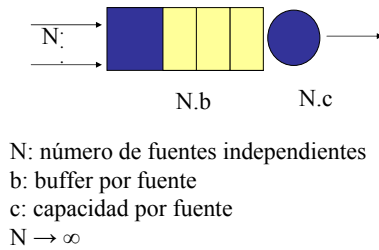


Figura 3.2: Régimen asintótico de muchas fuentes

La asintótica de muchas fuentes y buffer pequeño se analiza en la sección 3.5. Esta asintótica se diferencia de la anterior en que si bien la capacidad del enlace se escala proporcionalmente al crecimiento de las fuentes, el tamaño del buffer  $B(N)$  verifica que  $B(N)/N \rightarrow 0$  cuando la cantidad de fuentes  $N$  tiende a infinito (figura 3.3).

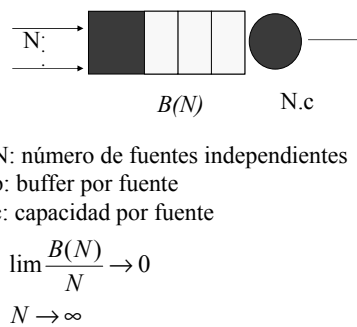


Figura 3.3: Régimen asintótico de muchas fuentes y buffer pequeño

Antes de proseguir en el análisis de este capítulo se debe comprender la razón por la cual se

plantean estas tres asintóticas diferentes en el análisis de desempeño de redes. Una de las razones por la que han surgido estas tres asintóticas diferentes es por el tipo de enlace al que se puede aplicar cada una de ellas. En particular, se analiza mediante asintóticas diferentes un enlace en el acceso a la red o un enlace del backbone de la red. En un backbone es mucho más razonable la hipótesis de “infinitos flujos” que la de “buffer grande”. Esto se debe a que en cada enlace confluyen una gran cantidad de agregados de flujos y el buffer por fuente en general es pequeño. El buffer en dichas redes se dimensiona sólo para atender la simultaneidad en la llegada de paquetes por las diferentes interfaces de entrada del enrutador. Al pasar por estos enrutadores una cantidad de flujos muy grande no es razonable dimensionar los buffers de los enlaces como para atender simultaneidad de ráfagas de los diferentes flujos. Por esta razón para el análisis de un enlace en el backbone es más adecuado utilizar la asintótica de muchas fuentes. La situación en un enlace de acceso es diferente. En este caso por el enrutador pasan pocos flujos y por lo tanto es más razonable darle un espacio grande de memoria a cada flujo para soportar ráfagas de paquetes en cada uno de ellos. Por este motivo el modelo de la asintótica de buffer grande es más adecuado cuando interesa analizar un enlace en el acceso a la red.

Por último, cabe preguntarse que agrega la asintótica de muchas fuentes y buffer pequeño a las analizadas antes. Esta asintótica al igual que la de muchas fuentes es adecuada para analizar enlaces de backbone donde como dijimos hay un número grande de flujos y el buffer por flujo es pequeño. Sin embargo, como se verá en este capítulo, en las dos asintóticas anteriores los resultados para extender el análisis de desempeño de un enlace a un camino son muy limitados. La principal causa de estas limitaciones es el buffer. Al esperar los paquetes en un buffer, se introduce dependencia en el tráfico que utiliza dicho enlace. Esta dependencia introducida en el tráfico a la salida de un enlace viola hipótesis necesarias para obtener los indicadores de desempeño tanto en el régimen de buffer grande como en el de muchas fuentes. Por lo tanto, a la salida de un enlace, el tráfico puede no verificar las mismas hipótesis que al ingreso y en ese caso no es posible aplicar el mismo método al siguiente enlace del camino. El régimen de muchas fuentes y buffer pequeño se puede pensar en el límite como un sistema sin buffer. Al “no existir buffer”, los problemas mencionados no se presentan de la misma forma. Aunque el tráfico se modifica al pasar por un enlace en el régimen de muchas fuentes y buffer pequeño, es posible establecer una recurrencia y extender los resultados de un enlace a un camino. En este último punto radica la importancia de este régimen para el análisis de desempeño de una red y también para el contexto de la tesis.

## 3.2. Resumen de la teoría de grandes desvíos

### 3.2.1. Introducción

Para analizar el desempeño utilizando las asintóticas anteriores, es necesario estudiar el comportamiento del proceso de ocupación de la cola en un enlace. La idea general del análisis de desempeño de una cola mediante la teoría de grandes desvíos, será encontrar la función que determina la velocidad con que tiende a cero la probabilidad de que la ocupación de la cola supere cierto umbral (que se denominará función velocidad del gran desvío). Para poder encontrar esta función y otras propiedades interesantes del gran desvío, como por ejemplo determinar cuál es el tiempo más probable en que se alcanza dicho umbral o cuál es la trayectoria mas probable con la

que se llega a superar dicho umbral, es necesario utilizar algunos resultados de la teoría de colas y de grandes desvíos que se reseñarán en esta sección.

En primer lugar, es necesario establecer una relación entre el tráfico de entrada y el proceso de servicio del enlace, con el proceso de ocupación de la cola. Los dos primeros, en general son conocidos o estimables a partir de mediciones. Por lo tanto, para conocer el tercero se debe establecer su relación con los dos primeros. Esta relación, bajo hipótesis muy generales, viene dada por la ecuación de Lidney y el teorema de Loynes. Estos dos resultados, se reseñarán en 3.2.2.

En 3.2.3, se dan algunas definiciones de la teoría de grandes desvíos, se define la función generatriz de momentos de una sucesión de variables aleatorias i.i.d. y su transformada de Fenchel-Legendre. Esta última juega un rol muy importante en la teoría de grandes desvíos, ya que será quien caracterice la función de velocidad del gran desvío del promedio de una sucesión de variables aleatorias. Esta relación entre la función velocidad de gran desvío y la transformada de Fenchel-Legendre se establece en el teorema de Cramer referido en 3.2.4. Si el tráfico de ingreso a la cola y el proceso de servicio fueran una sucesión de variables i.i.d. con el teorema de Cramer y el teorema de Loynes alcanzaría para encontrar la función velocidad del gran desvío de la ocupación de la cola del enlace. Sin embargo, en general no es así y el tráfico que ingresa a la cola tiene dependencias temporales. Por este motivo, es necesario extender el teorema de Cramer (para lo cual también es necesario extender la definición de la transformada de Fenchel-Legendre) para el caso en que la sucesión de variables del tráfico de ingreso no es i.i.d. porque existen dependencias temporales. Esto se establece en el teorema de Gartner-Ellis que se reseña en 3.2.5. En dicho teorema se establecen hipótesis adicionales al teorema de Cramer ya que se necesario definir el tipo de dependencia admisible para que la sucesión de variables aleatorias satisfaga un principio de gran desvío (LDP).

En varios casos en el análisis del comportamiento de una cola interesará no sólo observar con qué velocidad tiende a cero la probabilidad de que la ocupación de la cola supere cierto umbral, sino que también interesará analizar las trayectorias del proceso de ocupación de la cola (cómo se alcanza dicho desvío). Puede interesar analizar cuál es la trayectoria más probable que conduce a que la ocupación de la cola supere cierto umbral o cuál es la probabilidad de que una trayectoria esté a una distancia mayor que cierto umbral de la trayectoria más probable. Es necesario entonces, establecer un principio de gran desvío pero sobre el espacio de las trayectorias. Habitualmente se normalizan las trayectorias a algún espacio de funciones definido sobre el intervalo  $[0, 1]$  y se estudia la función velocidad del gran desvío de la trayectoria promedio. La función velocidad en este caso se establece en el teorema de Mogulskii que se reseña en 3.2.6.

Por último, en muchos casos se tiene la función velocidad de gran desvío para un proceso del tráfico de entrada a la cola y se desea encontrar la función velocidad de la ocupación de la cola o del tráfico a la salida del enlace. Se conoce que existe una función que relaciona estos procesos a partir del Teorema de Loynes. La pregunta es cómo se traduce esta relación a las funciones velocidad. El principio de contracción establece la relación entre las funciones velocidad de dos espacios vinculados a través de una función continua. El principio de contracción se reseña en 3.2.7.

### 3.2.2. Ecuación de Lindley y teorema de Loynes

Antes de comenzar con los resultados de grandes desvíos en esta sección se analizan algunos resultados clásicos de la teoría de colas, que serán utilizados en los apartados siguientes. El primer

resultado es conocido como la ecuación de Lindley. Sea una cola FIFO (first in first out), un proceso de arribo a la cola  $A(n)$ ,  $n \in \mathbb{Z}$ , y un proceso de servicio  $C(n)$ . Se denominará  $Q(n)$  al tamaño de la cola. Se define  $X(n) = A(n) - C(n)$ . En tiempo discreto  $A(n)$  es la cantidad de trabajo que arriba a la cola en el intervalo de tiempo  $(n, n + 1)$ ,  $C(n)$  es la cantidad de trabajo que puede ser procesado por el servicio en dicho intervalo y  $Q(n)$  representa la cantidad de trabajo que queda en la cola en el instante  $n$ . En tiempo continuo, los paquetes se numeran y  $C(n)$  representa el tiempo de servicio del paquete  $n$ ,  $A(n)$  la diferencia del tiempo entre arribos del paquete  $n$  y el  $n + 1$ ,  $Q(n)$  en este caso representa el tiempo de espera del paquete  $n$ .

La ecuación de Lindley gobierna el comportamiento de la cola:

$$Q(n + 1) = (Q(n) + X(n))^+ \quad (3.1)$$

donde  $Z^+$  es el máximo entre  $Z$  y 0.

De la ecuación de Lindley se puede escribir

$$\begin{aligned} Q(0) &= \max(X(-1) + Q(-1), 0) \\ &= \max(X(-1) + \max(X(-2) + Q(-2), 0), 0) \\ &= \max(X(-1), X(-1) + X(-2) + Q(-2), 0) \\ &= \max(X(-1), X(-1) + X(-2), \dots, \sum_{i=-n}^{i=-1} X(i) + Q(-n), 0) \end{aligned}$$

Si se define  $S_k = \sum_{i=-k}^{i=-1} X(i)$  con  $S_0 = 0$  entonces,

$$Q(0) = \max(S_0, S_1, \dots, S_n + Q(-n))$$

La pregunta que surge es si cuando  $n \rightarrow \infty$  este proceso tendrá límite. Es decir, en qué casos el trabajo en la cola “en régimen” tendrá una distribución estacionaria y cuándo ésta será independiente del estado inicial de la cola. Esto ha sido respondido por Loynes en 1962. A continuación se presenta una formulación del teorema de Loynes que no es exactamente la original, pero que será de utilidad en el presente trabajo.

**Teorema 3.1.** *Teorema de Loynes (ver su demostración por ejemplo en [10], pag. 74). Si se cumple que*

- *El proceso  $X$  es estacionario*
- *El proceso  $X$  es ergódico*
- *$E(X) < 0$*

*entonces para toda condición inicial de la cola  $Q$ , se verifica que*

$$\lim_{n \rightarrow \infty} P(Q(n) \leq x) = P(\sup_{n > 0} S_n \leq x) \quad (3.2)$$

*donde  $Q_0 = \sup_{n > 0} S_n$  es casi seguramente finita.*

### 3.2.3. Definiciones básicas de la teoría de grandes desvíos

En esta sección se brindarán las principales definiciones a usar en el resto de este trabajo. En la medida que sean necesarios para las aplicaciones a redes, se enunciarán los teoremas fundamentales de la teoría de grandes desvíos. La demostración de estos teoremas se puede encontrar en el libro de Dembo y Zeitouni [45], o en las referencias que se citarán.

Sea  $\mu_n$  una familia de medidas de probabilidad sobre un espacio  $(\chi, B)$

Una **función de velocidad** ( $RF$ ) es una función semicontinua inferiormente  $I : \chi \rightarrow [0, \infty)$  (las funciones semicontinuas por debajo son las funciones cuyos conjuntos de nivel son cerrados).

Se define  $\Psi_\alpha(I) = \{x : I(x) \leq \alpha\}$  el conjunto de nivel  $\alpha$  de  $I$ . Una función  $I$  es una **buena función de velocidad** si todos los conjuntos de nivel de  $I$  son conjuntos compactos de  $\chi$ .

**Definición 3.1.**  $\mu_n$  *satisface el principio de grandes desvíos (LDP) con función de velocidad*  $I$  si  $\forall \Gamma \in B$

$$- \inf_{x \in \Gamma^\circ} I(x) \leq \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(\Gamma) \leq \limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(\Gamma) \leq - \inf_{x \in \bar{\Gamma}} I(x) \quad (3.3)$$

con  $\Gamma^\circ$  el interior de  $\Gamma$  y  $\bar{\Gamma}$ , la clausura de  $\Gamma$

Cabe observar que si la función de velocidad es **buena**, el óptimo efectivamente se alcanza.

**Definición 3.2.**  $\mu_n$  *satisface el principio de las grandes desvíos débil con función de velocidad*  $I$ , si la cota superior de 3.3 se verifica para todo compacto y la cota inferior se alcanza para todo abierto.

**Definición 3.3.** Una familia de medidas  $\mu_n$  en  $\mathcal{X}$  es de decaimiento exponencial si para todo  $\alpha < \infty$  existe un compacto  $K_\alpha$  tal que

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(K_\alpha^C) \leq -\alpha$$

siendo  $K_\alpha^C$  el complemento de  $K_\alpha$ .

Si se cumple que la medida  $\mu_n$  tiene decaimiento exponencial entonces si cumple un LDP débil cumple también un LDP fuerte.

**Definición 3.4.** Sea  $X_1, \dots, X_n, \dots$  una sucesión de variables aleatorias i.i.d. en el espacio  $\mathbb{R}^d$  y sea  $\mu$  su ley de probabilidad.

Se llamará *función logarítmica generadora de momentos* de  $\mu$  a la función

$$\Lambda(\lambda) \triangleq \log(M(\lambda)) \triangleq \log(E(\exp(\langle \lambda, X_1 \rangle))) \quad (3.4)$$

donde  $\langle \lambda, X_1 \rangle$  es el producto interno habitual en  $\mathbb{R}^d$

**Definición 3.5.** Se llamará *transformada de Fenchel-Legendre* de  $\Lambda(\lambda)$  a la función

$$\Lambda^*(x) \triangleq \sup_{\lambda \in \mathbb{R}^d} \{\langle \lambda, x \rangle - \Lambda(\lambda)\} \quad (3.5)$$



### 3.2.4. Teorema de Cramér

**Teorema 3.2.** *Teorema de Cramér.* Sea  $\overline{S}_n$  con ley  $\mu_n$  definida de la siguiente manera

$$\overline{S}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

entonces  $\mu_n$  satisface un LDP débil con función de velocidad convexa  $\Lambda^*(x)$ .

Si además  $0 \in D_\Lambda^o$ , siendo de  $D_\Lambda^o = \{x : \Lambda(x) < \infty\}$  que se denomina dominio efectivo de  $\Lambda$ , entonces  $\mu_n$  satisface un LDP con “buena función de velocidad”  $\Lambda^*(x)$ .

**Observación 3.1.** La transformada de Fenchel-Legendre puede interpretarse como se muestra en la figura 3.4. La distancia máxima entre la recta  $\lambda x$  y la tangente a  $\Lambda(\lambda)$  con pendiente  $x$  es la transformada  $\Lambda^*(x)$ .

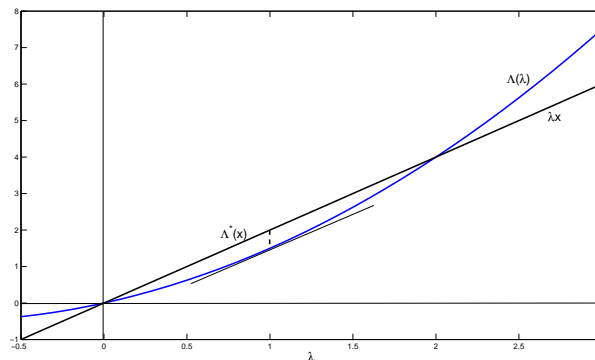


Figura 3.4: Transformada de Fenchel-Legendre

**Observación 3.2.** Para v.a. acotadas, es decir que existe algún  $K$  tal que

$$P(|X_i| > K) = 0,$$

el dominio efectivo de  $\Lambda$  es  $\mathbb{R}$ . Por ejemplo para el caso en que  $X$  toma valores entre 0 y 1 con probabilidad 1/2 se obtiene la siguiente función logaritmo de la generatriz de momentos

$$\Lambda(\lambda) = \log \left( \frac{\exp(\lambda) + 1}{2} \right)$$

y su transformada de Fenchel-Legendre

$$\begin{aligned} \Lambda^*(x) &= \log 2 + x \log(x) + (1-x) \log(1-x) \quad 0 \leq x \leq 1 \\ \Lambda^*(x) &= \infty \quad \text{en otro caso} \end{aligned}$$

**Observación 3.3.** *Variables Gaussianas independientes con media  $\mu$  y varianza  $\sigma^2$ . En este caso se obtiene la siguiente función logarítmica generatriz de momentos y su transformada de Fenchel-Legendre*

$$\begin{aligned}\Lambda(\lambda) &= \mu\lambda + \frac{\lambda^2\sigma^2}{2} \\ \Lambda^*(x) &= \frac{(x - \mu)^2}{2\sigma^2}\end{aligned}$$

*En este caso el dominio efectivo de  $\Lambda$  son todos los reales, al igual que en el caso anterior.*

**Observación 3.4.** *Variables exponenciales independientes de parámetro  $\mu$ . En este caso se obtiene la siguiente función logarítmica generatriz de momentos*

$$\begin{aligned}\Lambda(\lambda) &= \log\left(\frac{\mu}{\mu - \lambda}\right) & \lambda < \mu \\ \Lambda(\lambda) &= \infty & \lambda \geq \mu\end{aligned}$$

*y su transformada de Fenchel-Legendre*

$$\begin{aligned}\Lambda^*(x) &= \mu x - 1 - \log(\mu x) & x > 0 \\ \Lambda^*(x) &= \infty & x \leq 0\end{aligned}$$

*En este caso  $0 \in D_\Lambda^\circ$  pero a diferencia de los casos anteriores el dominio no es el conjunto de los reales.*

**Observación 3.5.** *Variables con distribución de Poisson con parámetro  $\mu$ . En este caso*

$$\Lambda(\lambda) = \mu(\exp(\lambda) - 1)$$

*y su transformada es:*

$$\begin{aligned}\Lambda^*(x) &= x \left( \log\left(\frac{x}{\mu}\right) - 1 \right) + \mu \text{ si } x > 0 \\ \Lambda^*(x) &= \mu \text{ si } x = 0 \\ \Lambda^*(x) &= \infty \text{ si } x < 0\end{aligned}$$

**Observación 3.6.** *Variables independientes con distribución de Cauchy. En este caso  $\Lambda(\lambda) = 0$  para  $\lambda = 0$  y vale  $\infty$  para todos los demás valores de  $\lambda$  y por lo tanto  $\Lambda^*(x) = 0$  para todo  $x$ .*

El teorema de Cramér se puede generalizar a variables no i.i.d. mediante el teorema de Gärtner-Ellis, que se verá a continuación

### 3.2.5. Teorema de Gärtner-Ellis

Sea una sucesión de variables aleatorias  $Z_n$  en  $\mathbb{R}^d$  con medida  $\mu_n$ . Se llamará

$$\Lambda_n(\lambda) = \log(E(\exp \langle \lambda, Z_n \rangle))$$

**Hipótesis 3.1.** Para todo  $\lambda$  perteneciente a  $\mathbb{R}^d$  existe como real extendido el siguiente límite

$$\Lambda(\lambda) = \lim_{n \rightarrow \infty} \frac{\Lambda_n(n\lambda)}{n} \quad (3.6)$$

y además el origen pertenece al interior del dominio efectivo de  $\Lambda$

**Definición 3.6.** Puntos expuestos de  $\Lambda^*$ . Un punto  $y \in \mathbb{R}^d$  es un punto expuesto de  $\Lambda^*$  si, para algún  $\theta$  perteneciente a  $\mathbb{R}^d$  y  $\forall x \neq y$ , se cumple que

$$\langle \theta, y \rangle - \Lambda^*(y) > \langle \theta, x \rangle - \Lambda^*(x)$$

$\theta$  es llamado el hiperplano expuesto de  $\Lambda^*$  y se llamará  $\mathcal{F}$  al conjunto de puntos expuestos de  $\Lambda^*$  (ver figura 3.5).

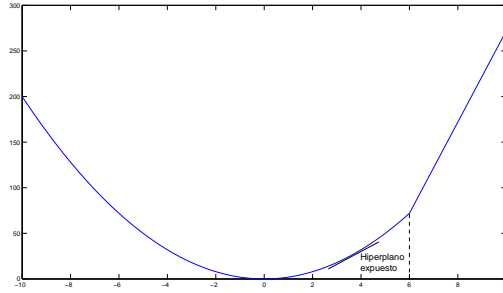


Figura 3.5: Puntos expuestos,  $\mathcal{F} = \{-10, 6\}$

**Definición 3.7.** Una función convexa  $\Lambda: \mathbb{R}^d \rightarrow (-\infty, \infty)$  es esencialmente suave si:

1.  $D_\Lambda^\circ \neq \emptyset$
2.  $\Lambda$  es diferenciable en el interior de su dominio
3.  $\Lambda$  es abrupta, esto quiere decir que  $\lim_{n \rightarrow \infty} |\nabla(\Lambda(\lambda_n))| = \infty$  cuando  $\{\lambda_n\}$  es una sucesión en el interior del dominio efectivo de  $\lambda$  convergente a un punto de la frontera de dicho dominio efectivo.

**Teorema 3.3.** Teorema de Gärtner-Ellis. Si se verifica la hipótesis 3.1 entonces se cumple que:

1. Para todo conjunto  $H$  cerrado y  $G$  abierto

$$\begin{aligned} \limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(H) &\leq - \inf_{x \in H} \Lambda^*(x) \\ - \inf_{x \in G \cap \mathcal{F}} \Lambda^*(x) &\leq \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(G) \end{aligned}$$

Siendo  $\mathcal{F}$  el conjunto de puntos expuestos de  $\Lambda^*$ .

2. Si además  $\Lambda$  es esencialmente suave y semicontinua inferior, se verifica el LDP con buena función de velocidad  $\Lambda^*(x)$

**Observación 3.7.** *Existen casos donde se verifica un LDP y sin embargo no se cumplen las hipótesis del teorema enunciado. Por ejemplo en el libro citado de Dembo y Zeitouni se señala el caso de una sucesión de variables aleatorias exponenciales de parámetro  $n$ .*

**Observación 3.8.** *Existen otros resultados donde no se puede aplicar el teorema ya que la función  $\Lambda$  no es diferenciable en algún punto del dominio.*

**Observación 3.9.** *Fracional Browniano. En este caso  $Z_n = \mu n + G_n$  donde  $G_n$  es una gaussiana con media 0 y varianza  $\sigma^2 n^{2H}$  con  $H \in (0, 1)$  el parámetro de Hurst. En este caso calculando  $\Lambda_n$  y realizando el límite se puede ver que este existe sólo si  $H \leq 1/2$ , es decir siempre que no se tengan dependencias largas. Este ejemplo permite ver que si bien no hay una hipótesis explícita sobre el tipo de dependencia de los procesos, las condiciones impuestas sobre la función  $\Lambda$  están relacionadas con la debilidad de su dependencia.*

### 3.2.6. Teorema de Mogulskii

Otro teorema clásico de la teoría de grandes desvíos es el teorema de Mogulskii. Existen generalizaciones de este teorema que se utilizan para demostrar algunos teoremas del régimen asintótico de buffer grande. La primera es debida a Dembo y Zajic [44] donde se generaliza el resultado de Mogulskii para el caso de variables no i.i.d. El segundo resultado debido a Ganesh y O’Connell [58] relaja ciertas hipótesis del teorema de Mogulskii y permite aplicar el principio de contracción para calcular la probabilidad de pérdida de un enlace. Los resultados de Crámer y Gärtner-Ellis permiten analizar el comportamiento en el límite de la ocurrencia de ciertos eventos “raros”. Lo que se pretende ahora es estudiar las trayectorias que conducen a esos eventos. Por ejemplo, el teorema de Gärtner-Ellis permitirá estudiar cuál es la probabilidad de que el tamaño de un buffer supere cierto valor (desborde del buffer). Pero para muchos problemas interesará también saber cómo se llega a que el buffer se desborde, o cuál es el tiempo en que es más probable que ocurra un desborde. Para responder a estas preguntas, es necesario estudiar grandes desvíos de las trayectorias de los procesos involucrados. El teorema de Mogulskii permitirá estudiar este tipo de problemas.

**Teorema 3.4.** *Sea  $X_1, X_2, \dots$  una sucesión de v.a. i.i.d. que toman valores en  $\mathbb{R}^d$  con función logarítmica generatriz de momentos definida como antes y verifica que  $\Lambda(\lambda) < \infty$  para todo  $\lambda \in \mathbb{R}^d$ . Se define  $\overline{S}_n$  como anteriormente y en este caso el teorema de Cramér dice cómo se comporta esta sucesión en el límite. Se define ahora la familia de v.a. indexadas por  $t$ :*

$$\widehat{S}_n(t) = \frac{1}{n} \sum_{i=1}^{[nt]} X_i \quad 0 \leq t \leq 1$$

donde  $[c]$  denota la parte entera de  $c$ . Sea  $\mu_n$  su ley de probabilidad.  $\mu_n$  satisface en el espacio  $L^\infty[0, 1]$  un LDP con buena función de velocidad

$$I(\phi) = \begin{cases} \int_0^1 \Lambda^*(\dot{\phi}(t)) dt, & \text{si } \phi \in \mathcal{AC}, \phi(0) = 0 \\ \infty, & \text{en otro caso} \end{cases} \quad (3.7)$$

donde  $\mathcal{AC}$  es el espacio de las funciones absolutamente continuas en  $[0, 1]$ .

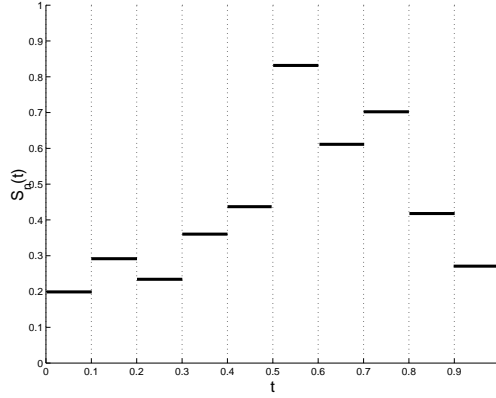


Figura 3.6:  $\widehat{S}_n(t)$  para  $n = 10$

En la figura 3.6 se muestra la interpretación de  $\widehat{S}_n(t)$  para  $n = 10$ .

Por otra parte, como en el espacio que se trabaja la distancia viene dada por la norma del supremo absoluto entre funciones, una interpretación del teorema de Mogulskii es que para un  $\varepsilon > 0$ ,

$$P(\|\widehat{S}_n - \phi\|_\infty < \varepsilon) \approx \exp\left(-n \inf_{\|\psi - \phi\|_\infty < \varepsilon} \int_0^1 \Lambda^*(\dot{\psi}(t)) dt\right)$$

Considerando  $\varepsilon > 0$  suficientemente pequeño se puede informalmente decir que la probabilidad de que el proceso  $\widehat{S}_n$  se encuentre “muy próximo” a una función  $\phi$  se puede aproximar por

$$\exp\left(-n \int_0^1 \Lambda^*(\dot{\phi}(t)) dt\right)$$

La interpretación gráfica se muestra en la figura 3.7.

### 3.2.7. Principio de Contracción

Este resultado será de mucha utilidad en el estudio de enlaces de comunicaciones pues permitirá, conocido un LDP para el tráfico de entrada, encontrar el LDP asociado a otras magnitudes que sean función de este tráfico, por ejemplo la distribución de ocupación de la cola de un enlace.

**Teorema 3.5.** Sean  $\mathcal{X}_1$  y  $\mathcal{X}_2$  dos espacios topológicos Hausdorff y sea  $f : \mathcal{X}_1 \rightarrow \mathcal{X}_2$  una función continua. Sea  $I$  una buena función de velocidad  $I : \mathcal{X}_1 \rightarrow [0, \infty]$

- para cada  $y \in \mathcal{X}_2$  se define

$$J(y) \triangleq \inf\{I(x) : x \in \mathcal{X}_1, y = f(x)\} \quad (3.8)$$

entonces  $J$  es una buena función de velocidad en  $\mathcal{X}_2$

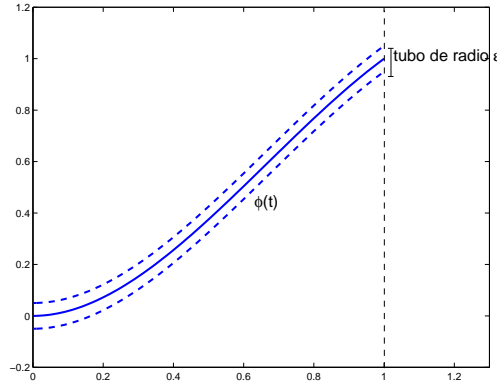


Figura 3.7: Un tubo de radio  $\varepsilon$

- Si  $I$  controla un LDP asociado con la familia de medidas de probabilidad  $\mu_n$  en  $\mathcal{X}_1$ , entonces  $J$  controla el LDP asociado con la correspondiente familia  $(\mu_n \diamond f^{-1})$  en  $\mathcal{X}_2$

Este teorema se apoya en que si la función es continua, entonces todo conjunto abierto(cerrado) del codominio se transforma en un conjunto abierto(cerrado) del dominio, entonces si vale un LDP en el dominio de la función, se extiende a un LDP en el codominio.

En las siguientes secciones se verán como se aplican estos resultados al análisis de desempeño de redes.

### 3.3. El régimen asintótico de buffer grande

#### 3.3.1. El análisis de un enlace

Sea  $S_n = X(-1) + \dots + X(-n)$ , con  $S_0 = 0$ . Como se vio según el teorema de Loynes, el tamaño de la cola en el enlace está dado por

$$Q_0 = \sup_{n \geq 0} S_n$$

**Teorema 3.6. Probabilidad de Pérdida en un enlace.** Se supone que el proceso  $X$  es estacionario y ergódico. Sea

$$\Lambda_n(\lambda) = \log(E(\exp(\lambda S_n/n)))$$

Se supone que el siguiente límite

$$\Lambda(\lambda) = \lim_{n \rightarrow \infty} \frac{1}{n} \log(E(\exp(\lambda S_n)))$$

existe para todo  $\lambda \in \mathbb{R}$  como un real extendido y que  $\Lambda(\lambda) < 0$  para algún  $\lambda > 0$ . Si  $\Lambda_n(\lambda) < \infty$

para todo  $\lambda > 0$  tal que  $\Lambda(\lambda) < 0$  entonces

$$\limsup_{q \rightarrow \infty} \frac{\log P(Q_0 \geq q)}{q} \leq -\delta$$

$$\delta = \sup(\lambda > 0 : \Lambda(\lambda) < 0)$$

Si la cota inferior del gran desvío

$$\limsup_{n \rightarrow \infty} \frac{\log P(S_n \geq xn)}{n} \geq \Lambda^*(x)$$

se verifica para todo  $x > 0$ , entonces

$$\liminf_{q \rightarrow \infty} \frac{\log P(Q_0 \geq q)}{q} \geq -\delta \quad (3.9)$$

Este Teorema se puede probar a partir del teorema de Gärtner-Ellis obteniendo la cota inferior y superior usando desigualdades clásicas de la teoría de probabilidad (ver por ejemplo [33]).

**Observación 3.10.** En la figura 3.8 se muestra una interpretación de este Teorema. Se recuerda que  $X(n) = A(n) - C(n)$ , es la diferencia entre cantidad de trabajo que arriba al enlace y la cantidad que es procesada por el servidor. Como  $\frac{d\Lambda}{d\lambda}(0) = \mu$  siendo  $\mu$  el valor medio de  $X$ , y  $\Lambda(0) = 0$ , la condición del Teorema que pide:  $\Lambda(\lambda) < 0$  para algún  $\lambda > 0$ , es equivalente a la condición de estabilidad de la cola del enlace (el valor medio de la entrada menor que el valor medio del servicio). Esto se aprecia en la figura 3.8. Además se puede ver que el valor  $\delta = \sup(\lambda > 0 : \Lambda(\lambda) < 0)$  que define la velocidad con que tiende a cero la probabilidad de pérdida, ocurre donde  $\Lambda(\delta) = 0$ . Si se considera el caso de un enlace de velocidad constante  $C$  entonces el punto de operación verifica que  $C = \frac{\Lambda_A(\delta)}{\delta}$  es decir que  $\frac{\Lambda_A(\delta)}{\delta}$  es la mínima capacidad necesaria del servidor para que esa fuente en un enlace con tamaño de buffer  $q$  tenga probabilidad de pérdidas:  $P(Q > q) \approx \exp(-q\delta)$ .

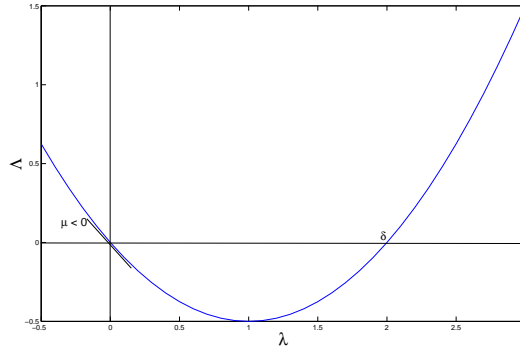


Figura 3.8: Teorema 3.6

**Observación 3.11.** *Ejemplo: tráfico gaussiano  $(\mu, \sigma^2)$  y servicio de capacidad  $C$  determinístico. En este caso  $\Lambda_A(\lambda) = \lambda\mu + \frac{\lambda^2\sigma^2}{2}$  y por lo tanto el valor  $\delta$  que verifica la ecuación  $C = \frac{\Lambda_A(\delta)}{\delta}$  es  $\delta = 2(C - \mu)/\sigma^2$ . Para una media  $\mu$  dada, se verifica que a mayor varianza menor es la velocidad con la que tienden a 0 las pérdidas. Por otra parte cuando la media se acerca a la capacidad del enlace la velocidad es cada vez menor.*

**Observación 3.12.** *Ejemplo: cola  $M/M/1$ . El tráfico que ingresa a la cola es Poisson de parámetro  $\mu_A$  y el servicio es exponencial de parámetro  $\mu_C$ . En este caso el punto de operación (ver ejemplo 3.6 para el caso poissoniano) queda:*

$$\begin{aligned}\Lambda(\lambda) &= \Lambda_A(\lambda) + \Lambda_C(\lambda) = 0 \\ \mu_A(\exp(\lambda) - 1) + \mu_C(\exp(-\lambda) - 1) &= 0 \\ \delta &= \log\left(\frac{\mu_C}{\mu_A}\right) \\ P(Q_0 \geq B) &\approx \left(\frac{\mu_A}{\mu_C}\right)^B\end{aligned}$$

### 3.3.2. Extensión de los resultados anteriores al análisis de una red

Para extender los resultados anteriores al caso de una red, se debe caracterizar el comportamiento del tráfico de salida de un enlace en este régimen. Existen varios resultados para casos particulares, ya sea del tráfico de entrada, del tipo de servicio, etc. Un resultado suficientemente general para la caracterización del tráfico de salida de un enlace en este régimen se debe a O'Connell [111]. En este trabajo se establece el siguiente teorema.

**Teorema 3.7.** *Sea un enlace que recibe  $d$  flujos de arribo  $\mathbf{A} = (A^1, \dots, A^d)$ , tiene un buffer FIFO infinito inicialmente vacío y un proceso de servicio  $C$ . Sea*

$$\mathbf{A}_n = \sum_1^n \mathbf{A}(n) \quad B_n = \sum_1^n C(n) \quad \mathbf{S}_n(t) = (\mathbf{A}_{[nt]}/n, B_{[nt]}/n)$$

Sea  $\mathbf{D}(\mathbf{n}) = (D^1(n), \dots, D^d(n))$  la cantidad de trabajo procesada de cada entrada en el tiempo  $n$ . Se supone que se verifica:

**Hipótesis 3.2.** *Para todo  $\lambda$ , el  $\sup_k E[\exp(\lambda(A(k) + C(k)))] < \infty$*

**Hipótesis 3.3.** *Para todo  $\lambda \in \mathbb{R}^{d+1}$ , el límite*

$$\Lambda(\lambda) = \lim_{n \rightarrow \infty} \frac{1}{n} \log E[\exp(\langle \lambda, S_n(1) \rangle)]$$

*existe como un real extendido y es finito en una vecindad del origen. Se verifica que  $\mathbf{S}_n(\mathbf{t})$  satisface un LDP en el espacio de las funciones  $L^\infty[0, 1]^{d+1}$  con buena función de velocidad dada por*

$$\begin{aligned}I_\infty(\phi) &= \int_0^1 \Lambda^*(\dot{\phi}) dt \quad \phi \in \mathcal{AC} \\ I_\infty(\phi) &= \infty \quad \text{en otro caso}\end{aligned}$$

donde  $\Lambda^*$  es la transformada de Fenchel-Legendre de  $\Lambda$



**Hipótesis 3.4.** *Los procesos de arribo y de servicio son asintóticamente independientes en el sentido de que*

$$\Lambda^*(\mathbf{a}, c) = \Lambda_a^*(\mathbf{a}) + \Lambda_b^*(c)$$

siendo  $\Lambda_a^*(\mathbf{a})$  y  $\Lambda_b^*(c)$ , las transformadas de Fenchel-Legendre del proceso de arribo y de servicio respectivamente.

Entonces  $\mathbf{D}_n/n$  satisface un LDP en  $\mathbb{R}^d$  con buena función de velocidad

$$\Lambda_d^*(z) = \inf\left\{\beta\Lambda_a^*(\mathbf{x}/\beta) + \sigma\Lambda_a^*\left(\frac{\mathbf{z}-\mathbf{x}}{\sigma}\right) + \beta\Lambda_b^*(c) + (1-\beta)\Lambda_b^*\left(\frac{z-x}{1-\beta}\right)\right\} \quad (3.10)$$

$$\beta, \sigma \in [0, 1], c \in \mathbb{R}, \beta + \sigma \leq 1, x \leq \beta c \quad (3.11)$$

Este resultado asume que la cola está inicialmente vacía, se llega a un resultado similar partiendo de la cola en estado estacionario. Estos resultados se basan en generalizaciones del teorema de Mogulskii visto antes.

En principio este resultado permitiría, aplicando los teoremas vistos para el análisis de un enlace, analizar una red. Como se conoce la función de velocidad de la salida, aplicando sucesivamente estos resultados sería posible conocer el desempeño de cualquier enlace. Sin embargo, para que esto sea cierto el tráfico de salida debe verificar las hipótesis del teorema anterior. En un trabajo de Ganesh y O’Connell [57] se prueba que en general no es cierto que el tráfico de salida verifique la hipótesis 3.3. Esta hipótesis habitualmente es referida como la propiedad de “geodésica lineal”. Esto se debe a la siguiente propiedad. Si se cumple la hipótesis 3.3 la trayectoria más probable que conduce al gran desvío  $\overline{S}_n \geq a$  es una trayectoria lineal. Se observa que  $\overline{S}_n = S_n(1)$  y por lo tanto

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log P(\overline{S}_n \geq a) = \lim_{n \rightarrow \infty} \frac{1}{n} \log P(\overline{S}_n = S_n(1) \geq a) \leq - \inf_{\phi(1) \geq a} \int_0^1 \Lambda^*(\dot{\phi}(t)) dt$$

Cómo  $\Lambda^*$  es convexa y  $\phi(0) = 0$  aplicamos la desigualdad de Jensen y se obtiene que

$$\int_0^1 \Lambda^*(\dot{\phi}(t)) dt \geq \Lambda^*(\phi(1))$$

y por lo tanto como  $\Lambda^*(a)$  es creciente para todo  $a > E[A]$  se cumple que

$$- \inf_{\phi(1) \geq a} \int_0^1 \Lambda^*(\dot{\phi}(t)) dt \geq \Lambda^*(a)$$

pero para  $\phi(t) = at$  se verifica que  $\int_0^1 \Lambda^*(\dot{\phi}(t)) dt = \Lambda^*(a)$  y por lo tanto  $\phi(t) = at$  es el camino más probable que conduce a el gran desvío.

La argumentación anterior se puede formalizar probando que la probabilidad de que la trayectoria no sea  $at$  dado que  $\overline{S}_n \geq a$  tiende a 0 cuando  $n$  tiende a infinito.

La propiedad de que la trayectoria “más probable” sea lineal se deriva de que la función de velocidad pueda ser expresada como la integral de una función convexa 3.3. En el trabajo de Ganesh y O’Connell se encuentran casos en los que esta propiedad no se preserva al pasar por un

enlace. Primero se estudia un contraejemplo para el caso  $d = 1$ , es decir cuando se tiene una sola clase de tráfico de entrada, y se ve que la salida no necesariamente cumple con la hipótesis referida.

En el trabajo del Dembo y Zajic referido antes, se prueba que si el proceso de tráfico cumple con algunas hipótesis de dependencia débil entonces se satisface la propiedad de “geodésica lineal”. El problema es que al pasar por un enlace la dependencia que se introduce puede llevar a que el tráfico a la salida pueda dejar de cumplir esta propiedad. Esta propiedad de geodésica lineal es característica de la asintótica de buffer grande. En el caso de la asintótica de muchas fuentes se verá que las trayectorias más probables de desborde del buffer no son necesariamente lineales.

### 3.4. El régimen asintótico de muchas fuentes

En la sección anterior se han reseñado los principales resultados del régimen de buffer grande. En esta sección, se estudia el desempeño de un enlace en el caso en que está alimentado por un número infinito de fuentes (escalando el buffer y la capacidad del enlace con el crecimiento de las fuentes).

Se considera un enlace al que arriban  $N$  fuentes independientes e idénticamente distribuidas y que tiene un tamaño de buffer igual a  $Nb$  y una capacidad  $Nc$ , siendo  $b$  y  $c$  el buffer y la capacidad por fuente (figura 3.2).

El régimen asintótico de muchas fuentes estudia este sistema cuando  $N \rightarrow \infty$ .

#### 3.4.1. Ancho de Banda Efectivo

Se utilizará la definición de ancho de banda efectivo (EB por su sigla en inglés) desarrollada por Kelly[80]. Sea  $X[0, t]$  la cantidad de trabajo acumulado que arriba desde una fuente en el intervalo  $[0, t]$ . Se asume que el proceso  $X[0, t]$  tiene incrementos estacionarios. Se define el ancho de banda efectivo como:

$$\alpha(s, t) = \frac{1}{st} \log(E(e^{sX[0,t]})) \quad (3.12)$$

La idea es que el valor de esta función para un cierto punto de operación  $(s^*, t^*)$  indicará la cantidad de ancho de banda mínimo del enlace que es necesario reservar para la fuente a los efectos de cumplir con los requerimientos de calidad de servicio. Se busca una función del tráfico de una fuente que dependiendo del contexto indique la cantidad de recursos que se debe reservar a la fuente. El contexto estará dado por el punto de operación que como se verá depende de la capacidad del enlace, del buffer, de las máximas pérdidas que se desean, y de otras fuentes que también alimenten el enlace. Una posibilidad es reservar el valor de pico de la fuente, en este caso no se tendrán pérdidas pero se estará desperdiciando capacidad. A partir del Teorema de Loynes se sabe que menos que la media no es posible reservar porque la cola del enlace no sería estable. Si se reserva un valor muy cercano a la media se tendrá un buen aprovechamiento de los recursos, pero las pérdidas probablemente estén por encima de las deseadas. Es deseable tener una función que caracterice a la fuente de tráfico y cumpla al menos con las siguientes condiciones:

- Sea mayor o igual que el valor medio de una fuente
- Sea menor o igual que el valor de pico de la fuente

- Si se tienen dos fuentes A y B estadísticamente independientes, el ancho de banda del total es deseable que sea función  $f(\alpha_A(s, t), \alpha_B(s, t))$  a los efectos de tener formas simples de agregación.

Estas condiciones son verificadas por la función  $\alpha(s, t)$  [33]. En particular el EB de la agregación de dos fuentes independientes es la suma de los EB de cada una de ellas. Pueden existir otras funciones “**ancho de banda efectivo**” que cumplan estas condiciones, pero esta definición tiene una interpretación muy natural y con algunas hipótesis adicionales se puede probar que es única. Como se vio para el régimen de buffer grande, si se supone que el servidor es determinístico de capacidad  $c$  por fuente, la probabilidad de pérdida en un enlace será (teorema 3.6)

$$\Lambda(s) = \lim_{t \rightarrow \infty} s\alpha(s, t) - cs$$

y en ese caso

$$\lim_{q \rightarrow \infty} \frac{\log P(Q_0 \geq q)}{q} \leq -\delta \text{ siendo}$$

$$\delta = \sup(s > 0 : \Lambda(s) \leq 0)$$

*es decir la solución de :*

$$\alpha(s, \infty) = c$$

De esta expresión se puede ver una primera interpretación. Si se desea en un enlace cuyo buffer tiene tamaño  $q$ , tener probabilidad de pérdida menor que  $\exp(-q\delta)$ , entonces la capacidad mínima que debe tener el enlace es el ancho de banda efectivo:  $\alpha(\delta, \infty)$ . Es decir que dependiendo del contexto, se obtiene un punto de operación  $s^*$  y el ancho de banda efectivo representa para ese contexto la capacidad mínima que se debe reservar a la fuente, para el nivel de pérdidas deseado.

La interpretación anterior es cierta solamente si el buffer es grande. Para tener una interpretación del ancho de banda efectivo en el caso de muchas fuentes se verán antes algunos resultados de este régimen.

### 3.4.2. Probabilidad de pérdida en un enlace

A partir de la expresión del ancho de banda efectivo, Wischik [158] demostró para el régimen asintótico de muchas fuentes una expresión que permite calcular la probabilidad de pérdida de un enlace. Esta expresión ya había sido encontrada por otros autores previamente [38], pero la demostración de Wischik facilita la interpretación de los resultados. Sea un enlace donde confluyen  $N$  fuentes de  $K$  tipos diferentes y donde las fuentes son i.i.d.. Se asigna una capacidad  $c$  y un tamaño de buffer  $b$  por fuente. Sea  $\alpha_j(s, t)$  el EB de una fuente del tipo  $j$ -ésimo que alimenta el enlace. Sea  $C = Nc$ ,  $B = Nb$  y  $\rho_j$  la proporción de fuentes de tipo  $j$ . Se verá que:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log(P(Q > B)) = -I \quad (3.13)$$

$$I = \inf_t \sup_s ((ct + b)s - st \sum_j^K \rho_j \alpha_j(s, t)) \quad (3.14)$$

Esto permite estimar la probabilidad de pérdidas en un enlace, donde se conoce el ancho de banda de los flujos agregados que arriban a él a través de la siguiente expresión

$$P(Q > B) \approx \exp(\inf_t \sup_s ((Ct + B)s - st \sum_j^K N \rho_j \alpha_j(s, t))) \quad (3.15)$$

es decir,

$$P(Q > B) \approx \exp(-NI) \quad (3.16)$$

La forma de calcular la probabilidad de pérdida en un enlace es resolver la doble optimización de la ecuación 3.14, de donde se obtiene un punto de operación  $(s^*, t^*)$  y la probabilidad de pérdida. El parámetro  $t$  se le llama parámetro temporal y  $t^*$  como se verá representa el tiempo más probable al desborde del buffer del enlace. El parámetro  $s$  se denomina parámetro espacial y  $s^*$  indica el grado de multiplexado estadístico del enlace.

En el contexto del trabajo de Wischik  $t$  será una variable que toma valores sobre el conjunto de los naturales. Sea  $\mathcal{X}$ , el espacio de los procesos a valores reales indexados por  $t$ . Un proceso de  $\mathcal{X}$  se notará como  $\mathbf{x}(0, \infty)$  y cuando se trunca al conjunto  $s, s + 1, \dots, t$  se notará  $\mathbf{x}(s, t)$  para  $s < t$ . Se notará  $\mathbf{1}$  al proceso que toma valores 1 constantes en todos los instantes de tiempo. Se trabajará con una sucesión de procesos  $(\mathbf{X}^N : N = 1, \dots, \infty)$ . En el trabajo de Wischik  $\mathbf{X}^N$  es vista como el promedio de  $N$  fuentes independientes e idénticamente distribuidas. El objetivo es encontrar un LDP para  $\mathbf{X}^N$  y luego aplicar el principio de contracción para obtener un LDP para la probabilidad de pérdida. Los resultados principales a los que llega Wischik para el análisis del gran desvío de la probabilidad de desborde de un enlace se resumen en los siguientes teoremas.

**Teorema 3.8. Fórmula infsup**

*Si se cumplen las siguiente hipótesis:*

**Hipótesis 3.5.** *Sea la función logarítmica generadora de momentos  $\Lambda_t^N$  para  $\mathbf{s} \in \mathbb{R}^t$  definida por*

$$\Lambda_t^N(\mathbf{s}) = \frac{1}{N} \log E(\exp(N \langle \mathbf{s}, \mathbf{X}^N(0, t] \rangle))$$

*Se asume que para cada  $t$  y  $\mathbf{s}$  la función límite generadora de momentos*

$$\Lambda_t(\mathbf{s}) = \lim_{N \rightarrow \infty} \Lambda_t^N(\mathbf{s})$$

*existe como un real extendido y el origen pertenece al interior del dominio efectivo de  $\Lambda_t(\mathbf{s})$ . Se asume también que  $\Lambda_t(\mathbf{s})$  es una función esencialmente suave.*

**Hipótesis 3.6.** *Una función de escala es una función  $v : \mathbb{N} \rightarrow \mathbb{R}$  para la cual  $v(t)/\log(t) \rightarrow \infty$ . Para alguna función de es escala  $v$  se define la función:*

$$\Lambda_t^N(\mathbf{s}) = \frac{1}{v(t)} \Lambda_t^N(\mathbf{1}sv(t)/t)$$

*para  $\mathbf{s} \in \mathbb{R}$ . Por la hipótesis 3.5, existe una vecindad abierta del origen en la cual el límite*

$$\Lambda_t(\mathbf{s}) = \lim_{N \rightarrow \infty} \Lambda_t^N(\mathbf{s})$$

existe. Se asume que existe una vecindad abierta del origen en la cual este límite y el límite

$$\Lambda(\mathbf{s}) = \lim_{t \rightarrow \infty} \Lambda_t(\mathbf{s})$$

existen uniformemente en  $s$ . Se asume que para  $s$  en alguna vecindad abierta del origen, el límite

$$\sqrt{\frac{v(t)}{\log t}} (\Lambda_t^N(s) - \Lambda(s)) \rightarrow 0$$

uniformemente en  $s$  cuando  $t, N \rightarrow \infty$ .

**Hipótesis 3.7.** Se asume que  $\Lambda'_t(s\mathbf{1}) < Ct$  en  $s = 0$  para todo  $t$ .

Entonces  $Q(\mathbf{X}^N)$  satisface un LDP con buena función de velocidad

$$I(b) = \inf_t \sup_s (s(b + Ct) - \Lambda_t(s\mathbf{1})) \quad (3.17)$$

**Teorema 3.9.** Si  $I(b)$  es finita, entonces la escala de tiempo óptima  $t^*$  y el camino óptimo  $\mathbf{x}^*(0, t^*)$  son ambos alcanzados y si el óptimo del parámetro espacial  $s^*$  es alcanzado entonces

$$\mathbf{x}^*(0, t^*) = \nabla \Lambda_{t^*}(s^*\mathbf{1}) \quad (3.18)$$

El resultado del Teorema 3.8 permite estimar la probabilidad de pérdida en un enlace por

$$P(Q > B) \approx \exp^{-NI}$$

Es decir que si es posible conocer o estimar para el tráfico de entrada  $\Lambda_t(s\mathbf{1})$ , se tendrá una aproximación de la probabilidad de pérdida en el enlace. Por otra parte el teorema 3.9 dice cómo será en este caso la “trayectoria más probable” que conduce al desborde del buffer. En el caso de la asintótica de “buffer grande” se vio que esta trayectoria era lineal, sin embargo la expresión 3.18, permite verificar para casos simples que en el caso de la asintótica de muchas fuentes el camino más probable  $\nabla \Lambda_{t^*}(s^*\mathbf{1})$  no necesariamente es el lineal. Por otro lado se puede apreciar de esta expresión la interpretación de  $t^*$ , es el tiempo en que es más probable que el buffer se llene desde cero hasta un nivel  $b$ . El parámetro  $s^*$  indica el grado de multiplexado estadístico. Si el valor de  $s^*$  es próximo a cero se estará reservando un EB próximo a la media por lo cual se tendrá un aprovechamiento alto del multiplexado estadístico, mientras que si  $s^* \rightarrow \infty$ , se estará reservando un EB próximo al valor de pico y por lo tanto aprovechando poco el multiplexado estadístico.

Un último lema importante que deduce Wischik en este trabajo es que el evento de que la cola no esté vacía en el régimen asintótico de muchas fuentes tiene una cota superior de gran desvío dada por :

$$I = \sup_s (sC - \Lambda_1(s\mathbf{1})) \quad (3.19)$$

Es decir que la probabilidad de que la cola no esté vacía en el régimen asintótico de muchas fuentes tiende a cero y con velocidad mayor que 3.19. Este resultado se utilizará en la sección siguiente. Eun y Shroff [52] demuestran para tiempo discreto y continuo (con hipótesis adicionales) que  $\lim_{N \rightarrow \infty} Q^N(t) \rightarrow 0$  casi seguramente.

Wischik también observa que el resultado de los teoremas 3.8 y 3.9 se verifican para el caso de un enlace con buffer  $B$  finito.

### 3.4.3. De un enlace al análisis de una red, la propuesta de Wischik

El objetivo final de este estudio es el análisis de una red de extremo a extremo. Se trata de analizar el desempeño de toda la red y no sólo de un enlace. Un primer paso es el análisis del proceso de salida de un enlace. Se buscará caracterizar la salida de un enlace a partir del tráfico de entrada. Si además el tráfico de salida verifica las mismas hipótesis que el de entrada, entonces será posible aplicar el mismo método de análisis que al primer nodo a los nodos siguientes de la red. Wischik demuestra en [157] que en el régimen asintótico de muchas fuentes, la función ancho de banda efectivo  $\alpha(s, t)$  de **un flujo** del tráfico de salida en el límite es igual a la función ancho de banda efectivo de **un flujo** de la entrada. Se verá a continuación este resultado con más detalle. En el modelo asintótico de muchas fuentes de Wischik el proceso  $\mathbf{X}^N$  es el promedio de  $N$  fuentes independientes e idénticamente distribuidas. Se considera ahora una de esas  $N$  fuentes de entrada que se notará  $\mathbf{X}^{(N)}$ . De la misma forma se identificará una de las  $N$  fuentes de la salida por  $\tilde{\mathbf{X}}^{(N)}$ . La función generadora de momentos  $\Lambda_t^N$  para la entrada agregada es

$$\Lambda_t^N(s) = \log E(\exp(\langle s, \mathbf{X}^{(N)} \rangle))$$

y para el agregado de copias independientes de una salida típica será

$$\tilde{\Lambda}_t^N(s) = \log E(\exp(\langle s, \tilde{\mathbf{X}}^{(N)} \rangle))$$

**Teorema 3.10. Caracterización de la salida en tiempo finito** *Se asume que  $\mathbf{X}^{(N)}$  satisface las hipótesis 3.5 y 3.6 y que dicho proceso es estacionario y con media estrictamente menor que  $C$ . Llamando a la función límite generadora de momentos de la entrada  $\lim_{N \rightarrow \infty} \log E(\langle s, \mathbf{X}^{(N)}(0, t] \rangle) = \Lambda_t(s)$ , entonces el proceso de salida  $\tilde{\mathbf{X}}^{(N)}$  satisface la hipótesis 3.5 y con la misma función límite generatriz de momentos que  $\mathbf{X}^{(N)}$ .*

La demostración de este teorema se basa en que la  $P(Q_0^N > 0) \rightarrow 0$  y por lo tanto la probabilidad de que los procesos de entrada y salida en un intervalo de tiempo fijo sean iguales tiende a 1. La demostración se hace en la hipótesis de que el buffer es finito para poder acotar la salida acumulada por la entrada acumulada  $\tilde{X}^{(N)}(0, t] \leq X^{(N)}(0, t] + \lfloor B/C \rfloor$ . Esto se usa para ver que  $\exp(s \cdot \tilde{\mathbf{X}}^{(N)})$  es uniformemente integrable. De estas dos conclusiones se ve que  $E(\exp(s \tilde{X}^{(N)}(0, t])) - E(\exp(s X^{(N)}(0, t])) \rightarrow 0$

Si bien el resultado anterior es importante, no es suficiente para poder aplicar el principio de contracción a la salida. Es necesario que sea válido el teorema 3.8 para lo cual la salida debe verificar la hipótesis 3.6. El problema es que esta hipótesis no se puede verificar en el espacio  $(\mathcal{X}_\mu, \|\cdot\|)$ . Wischik demuestra que esta hipótesis se cumple usando una topología más débil (se notará  $wq$ ) dada por la métrica

$$d(\mathbf{x}, \mathbf{y}) = |Q(\mathbf{x}) - Q(\mathbf{y})| + \sum_{t=1}^{\infty} \frac{1 \wedge |x_t - y_t|}{2^t} \quad (3.20)$$

**Teorema 3.11. Caracterización de la salida en tiempo largos** *Se asume que  $\mathbf{X}^{(N)}$  satisface las hipótesis 3.5 y 3.6, y que dicho proceso es estacionario y con media estrictamente menor que  $C$ , entonces el proceso de salida  $\tilde{\mathbf{X}}^{(N)}$  satisface un LDP en el espacio  $(\mathcal{X}_\mu, wq)$  ( $\mu$  mayor que la media del proceso) con la misma buena función de velocidad que el proceso de entrada*

La demostración de este teorema es una aplicación del inverso del principio de contracción probando que  $\tilde{\mathbf{X}}^{(N)}$  es de decaimiento exponencial en el espacio  $(\mathcal{X}_\mu, wq)$ .

Lo que demuestra Wischik es que si se agrega un flujo de la salida de un enlace, con  $N$  flujos i.i.d. con él, el ancho de banda efectivo de este agregado cuando  $N$  tiende a infinito es igual al ancho de banda efectivo del agregado de  $N$  fuentes i.i.d. de la entrada al enlace. Esto permite estudiar cierto tipo de redes por ejemplo redes con topologías donde a la salida de cada enlace de nivel  $k$  se toma un flujo y se lo agrega con  $N - 1$  flujos del mismo tipo provenientes de otros enlaces de ese nivel, y ese agregado independiente se usa como entrada a un enlace de nivel  $k + 1$ . Este resultado no implica que el ancho de banda efectivo de la salida agregada sea igual al ancho de banda efectivo de la entrada agregada, ya que los flujos a la salida de un enlace presentan dependencia por pasar por el buffer. Por este motivo la potencialidad de estos resultados en su aplicación a una red son limitados. En la siguiente sección se analizará una asintótica que brindará resultados mas interesantes para este fin.

### 3.5. La asintótica de muchas fuentes y buffer pequeño

En esta sección se analizará un trabajo de Ozturk, Mazumdar y Likhanov [116]. A partir de este trabajo, se realizará en el siguiente capítulo un estudio que permite simplificar el análisis de desempeño planteado por estos autores y encontrar condiciones bajo las cuales este análisis simplificado da resultados exactos. Además se encontrarán cotas del error para los casos donde este análisis no de resultados exactos.

En el trabajo de Ozturk et al. se estudia una red con enlaces con “muchas fuentes”, donde la capacidad del enlace se escala proporcionalmente al crecimiento de las fuentes, pero el tamaño del buffer verifica que  $B(N)/N \rightarrow 0$  cuando la cantidad de fuentes  $N$  tiende a infinito.

#### 3.5.1. Análisis de desempeño de una red alimentada por muchos flujos y buffer pequeño

A continuación se resumen los principales resultados del trabajo de Ozturk et al. [116].

Se considera una red en tiempo discreto, la capacidad del nodo  $k$  es  $NC_k$  y el tráfico que no puede ser servido se almacena en un buffer FIFO de tamaño  $B_k(N)$  con  $B_k(N)/N \rightarrow 0$  con  $N \rightarrow \infty$ . El tráfico que no puede ser servido, si el buffer está lleno se descarta. A la red de  $K$  enlaces arriba tráfico de  $M$  tipos de fuentes. El tráfico de cada clase (tipo) es independiente del de todas las otras clases, pero no se asume por ahora independencia dentro de cada clase.

$X_{k,t}^{m,N}$  es la cantidad de trabajo que arriba de  $N$  fuentes de tipo  $m$ , al enlace  $k$ , en el instante  $t$ .

Cuando se nota  $X_t^{m,N}$  se hace referencia al tráfico que arriba a la red de la clase  $m$  en el instante  $t$ .

El proceso de tráfico de entrada a la red para todas las clases es estacionario y ergódico.

Se llamará  $X_k^{m,N}(0, t)$ , al trabajo acumulado de  $N$  fuentes de tipo  $m$  en el enlace  $k$  en el intervalo  $(0, t)$ . Se nota  $X_0^{m,N}(0, t)$  cuando corresponde al tráfico de entrada a la red.

Sea  $\mu_m^N = E(X_0^{m,N})/N$ .

Se asumirá que  $\lim_{N \rightarrow \infty} \mu_m^N = \mu_m$ . Se asumirá que  $X^{m,N}(0, t)/N$  satisface un LDP con buena

función de velocidad  $I_t^{X^m}(x)$ :

$$\begin{aligned} - \inf_{x \in \Gamma^o} I_t^{X^m}(x) &\leq \liminf_{N \rightarrow \infty} \frac{1}{N} \log P(X^{m,N}(0, t)/N \in \Gamma) \\ &\leq \limsup_{N \rightarrow \infty} \frac{1}{N} \log P(X^{m,N}(0, t)/N \in \Gamma) \leq - \inf_{x \in \bar{\Gamma}} I_t^{X^m}(x) \end{aligned} \quad (3.21)$$

donde  $\Gamma \in \mathbb{R}$  es un conjunto de Borel con interior  $\Gamma^o$  y clausura  $\bar{\Gamma}$  e  $I_t^{X^m}(x) : \mathbb{R} \rightarrow [0, \infty)$  es una función continua con conjuntos de nivel compactos. No se hacen hipótesis sobre la independencia y equidistribución de las fuentes dentro de cada clase. Si las  $X^{m,N}(0, t)$  son i.i.d. como en el régimen asintótico de muchas fuentes, el LDP 3.21 vale por Crámer, pero esto no es una exigencia en este trabajo.

Se asume también una condición técnica que se verifica para los modelos de tráfico de interés:

Para todo  $m$  y  $a > \mu_m$ ,  $\liminf_{t \rightarrow \infty} \frac{I_t^{X^m}(at)}{\log t} > 0$

Se asume que la red tiene un ruteo fijo y sin bucles. El tipo de tráfico  $m$  tiene un camino en la red representado por el vector  $\mathbf{k}^m = (k_1^m, \dots, k_{l_m}^m)$ , donde  $k_i^m \in (1, \dots, K)$ . El conjunto  $\mathcal{M}_k = \{m : k_i^m = k, 1 \leq i \leq l_m\}$  denotará los tipos de tráfico que pasan a través del nodo  $k$ . Se asume que

$$\sum_{m \in \mathcal{M}_k} \mu_m < C_k$$

es decir que todas las colas son estables.

**Teorema 3.12.** *Existe una función continua  $g_k^m : \mathbb{R}^M \rightarrow \mathbb{R}$  que relaciona la velocidad instantánea del tráfico de entrada al nodo  $k$ , para la clase de tráfico  $m$ , con las velocidades instantáneas de todos los tipos de tráfico externos de entrada a la red tal que:*

$$X_{k,0}^{m,N}/N = g_k^m(X_0^{1,N}/N, \dots, X_0^{M,N}/N) + o(1) \quad (3.22)$$

bajo la hipótesis de estabilidad 3.22, la probabilidad de desborde del buffer viene dada por:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\text{desborde en el nodo } k) = -\mathbf{I}_k = \quad (3.23)$$

$$- \inf \left\{ \sum_{m=1}^M I_1^{X^m}(x_m) : x = (x_m) \in \mathbb{R}^M, \sum_{m=1}^M g_k^m(x) > C_k \right\} \quad (3.24)$$

Se verá a continuación un esbozo de la demostración de este teorema. Este teorema se basa en dos puntos. El primero es encontrar la forma de la función  $g_k^m$  y ver que siempre existe. El segundo es usando esta función aplicar el principio de contracción para obtener el LDP sobre la probabilidad de pérdida a partir del LDP de las entradas a la red. Para obtener la función  $g_k^m$ , se analiza un enlace  $j$  cualquiera de la red y se llamará  $X_{j,0}^{m,N}$  ( $Y_{j,0}^{m,N}$ ) al tráfico instantáneo de tipo  $m$  entrante (saliente) al enlace  $j$ . Se define para  $n = 1, \dots, M$  la función

$$f_n(x_1, \dots, x_M, y) = \frac{x_n y}{\max(\sum_{i=1}^M x_i, y)} \quad (3.25)$$



Como el buffer es FIFO, si no hay otras hipótesis sobre diferenciación entre las clases, la capacidad del enlace se llena en proporción a la cantidad de tráfico que arriba de cada clase. Si el buffer en el nodo  $j$  está vacío en  $t = -1$ , entonces la cantidad de capacidad usada por  $X_{j,0}^{n,N}$  será igual a  $f_n(x_1, \dots, x_M, NC_j)$ . Si el buffer no está vacío, la capacidad “equivalente” mínima disponible en ese instante será la capacidad del enlace menos el tamaño del buffer. Por lo tanto el peor caso para la cantidad de tráfico de la clase  $n$  que sale del enlace  $j$  en un cierto instante es cuando el buffer está lleno pero no tiene tráfico de esta clase:

$$Y_{j,0}^{n,N} \geq f_n(X_{j,0}^{1,N}, \dots, X_{j,0}^{M,N}, NC_j - B_j(N)) \quad (3.26)$$

$$\geq f_n(X_{j,0}^{1,N}, \dots, X_{j,0}^{M,N}, NC_j) - B_j(N) \quad (3.27)$$

donde la última desigualdad se puede verificar simplemente realizando operaciones. Por otra parte, la salida verifica que

$$Y_{j,0}^{n,N} \leq f_n(X_{j,0}^{1,N}, \dots, X_{j,0}^{M,N}, NC_j) + B_j(N) \quad (3.28)$$

ya que el peor caso es que el buffer estuviera lleno sólo de tráfico de tipo  $n$  y el resto se cubriera en proporción a la cantidad instantánea del tráfico de tipo  $n$  sobre el total de tráfico instantáneo arribado. Relacionando ambas ecuaciones, dividiendo entre  $N$ , se llega a que:

$$Y_{j,0}^{n,N}/N = f_n(X_{j,0}^{1,N}/N, \dots, X_{j,0}^{M,N}/N, C_j) + o(1) \quad (3.29)$$

Para obtener la función que relaciona el tráfico de entrada a un nodo  $k$   $X_{k,0}^{m,N}$  y el tráfico externo de entrada a la red  $X_0^{m,N}$ , si la red es feed-forward, esto se deriva de aplicar sucesivamente de afuera hacia adentro el resultado anterior. Las redes que interesan a los efectos prácticos son en general feed-forward. Ozturk et al. demuestran la existencia de tal función aún para redes no necesariamente feed-forward. No se verá el detalle de esa parte de la demostración y se utilizará el resultado para redes feed-forward. Dada la existencia de la función  $g_k^m$  se deducirá el LDP para la probabilidad de pérdida. Como notan los autores, la función  $g_k^m$  en general no es fácil de encontrar, pero si la red es feed-forward se puede calcular componiendo las  $f_n$ , en otros casos es aún más complicada de calcular.

Se observa que del resultado anterior y aplicando el principio de contracción se obtiene el LDP para  $X_{k,0}^{m,N}$ , la cual satisface un LDP con buena función de velocidad :

$$I_1^{X_k^m}(y) = - \inf \left( \sum_{n=1}^M I_1^{X^n}(x_n) : x = (x_n) \in \mathbb{R}^M, g_k^m(x_1, \dots, x_M) = y \right) \quad (3.30)$$

Observar que entre clases hay independencia.

Se considera ahora la entrada total al nodo  $k$ , que se notará  $Z_k^N$  y se define  $g_k = \sum_{m \in \mathcal{M}_k} g_k^m(x)$ , entonces  $Z_{k,0}^N/N = g_k(X_0^1/N, \dots, X_0^M/N)$  y satisface un LDP con buena función de velocidad:

$$I_1^{Z_k}(y) = - \inf \left( \sum_{n=1}^M I_1^{X^n}(x_n) : x = (x_n) \in \mathbb{R}^M, g_k(x_1, \dots, x_M) = y \right) \quad (3.31)$$

A partir de esta ecuación, y operando se obtiene la cota superior e inferior de la probabilidad de pérdida, llegando al resultado del teorema 3.12.

El resultado 3.31 muestra que en una red con buffers pequeños, para un tipo de tráfico  $m$ , la función de velocidad del tráfico de ese tipo de fuente a la entrada a cualquier nodo interno de la red depende no sólo de la función de velocidad externa de esa fuente, sino también de todas las demás fuentes, además de los parámetros y topología de la red. Es decir que en el caso general, el ancho de banda efectivo de un agregado de fuentes de un cierto tipo aún en este caso de buffers pequeños no se preserva, sino que al atravesar los nodos se modifica porque se genera dependencia con las demás fuentes.

Por último Ozturk et al. definen el ratio de pérdidas de una clase de fuentes al atravesar la red y analizan la región de aceptación para asegurar que el ratio de pérdidas de cada clase de fuentes esté por debajo de los niveles de QoS requeridos para cada una.

Para cada tráfico de entrada de tipo  $m$  se define  $\mathbf{L}^{m,N}$  llamado ratio total de pérdidas. Se define como el ratio entre el valor esperado de bits perdidos en todos los nodos a lo largo de su ruta y la media en bits del tráfico de entrada de ese tipo.

Sea  $r^m$  el conjunto de nodos por los que pasa la ruta del tráfico  $m$ . Entonces

$$\mathbf{L}^{m,N} = \sum_{k \in r^m} \frac{L_k^{m,N}}{E[X_t^{m,N}]} \quad (3.32)$$

donde  $L_k^{m,N}$  es el valor esperado de bits perdidos en el nodo  $k$ , para el tráfico  $m$  definido como

$$L_k^{m,N} = E[(X_{k,t}^{m,N} + Q_{k,t-1}^{m,N} - C_{k,t}^{m,N})^+ - Q_{k,t}^{m,N}] \quad (3.33)$$

A partir de esta expresión operando se demuestra que

**Teorema 3.13.**

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbf{L}^{m,N} = - \min_{k \in r^m} \mathbf{I}_k \quad (3.34)$$

Este teorema permite conociendo la función velocidad de la probabilidad de pérdida en cada nodo de la red conocer la función velocidad del ratio de pérdida de punta a punta de cada tipo de tráfico.

El último resultado de este trabajo que se utilizará en el capítulo próximo se refiere a la región de aceptación de un conjunto de flujos en la red. La red aceptará una cantidad de flujos que arriban si cumplen con los requerimientos de calidad de servicio.

Se asume que  $X^{m,N}$  es la suma de  $N$  procesos i.i.d. En esta hipótesis se define la región de aceptación que se notará  $\mathcal{D}$ . Esta región corresponde a la colección  $\{n_m\}_{m=1}^M$  de fuentes que cuando están presentes en la red resulta en que cada clase cumple sus requerimientos de QoS sobre el ratio de pérdidas:

$$\mathcal{D} = \{(n_m), m = 1, \dots, M : \lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbf{L}^{m,N} < -\gamma_m\} \quad (3.35)$$

Mazumdar establece una condición sobre la región de aceptación que se utilizará en el próximo capítulo:

**Teorema 3.14.** *Sea  $\mathcal{D}$  la región de aceptación para  $(n_m)$  definida antes. Se considera la red ficticia donde  $X^{m,N}$  llega a cada nodo de su camino sin ser afectado por los nodos anteriores a ese nodo y sea  $\overline{\mathcal{D}}$ , la región de aceptación en este caso. Entonces,*

$$\overline{\mathcal{D}} \subseteq \mathcal{D} \quad (3.36)$$

Este Teorema es una de las bases de esta tesis ya que permite trabajar con la red ficticia y estar seguros que no se subestimarán el ratio de pérdidas. Esto permite tomar decisiones basados en el análisis de la red ficticia y estar seguros que se verifican los requerimientos de QoS de la red real. Puede suceder que se sobrestimen las pérdidas. Por esta razón se analizará con detalle en el capítulo siguiente la red ficticia y se encontrarán condiciones bajo las cuales el análisis sobre la red ficticia es exacto. Este estudio permitirá también entender cuando el análisis sobre la red ficticia no es exacto de qué depende la magnitud del error y encontrar cotas para el mismo.

### 3.6. Conclusiones

En este capítulo, se realizó una reseña de los principales resultados de la teoría de grandes desvíos que se utilizan en el análisis de redes. Posteriormente, estudiaron diferentes asintóticas de la teoría de grandes desvíos para el análisis de desempeño de una red de telecomunicaciones. Dos de estas asintóticas (muchas fuentes y muchas fuentes con buffer pequeño) se pueden aplicar al análisis de un enlace del corazón de la red. La otra asintótica (la asintótica de buffer grande) se puede aplicar fundamentalmente a un enlace en la red de acceso. En cuanto a la aplicación de estas técnicas al análisis de desempeño de un camino y no sólo de un enlace, el resultado más general se obtiene en la asintótica de muchas fuentes y buffer pequeño. En este contexto surge el análisis mediante el método de la red ficticia, que se estudiará con detalle en el próximo capítulo donde se presentarán algunos resultados de la tesis en esta área.

## Capítulo 4

# Análisis mediante el método de la red ficticia

### 4.1. El análisis mediante la red ficticia

En el capítulo anterior se analizó el trabajo de Ozturk et al. [116]. Este trabajo permite analizar la probabilidad de desborde de cualquier enlace en una red y calcular el ratio de pérdidas en un camino. Sin embargo, para poder hacerlo es necesario calcular recursivamente ciertas funciones en cada enlace de la red y esto representa una dificultad computacional para aplicar este método en línea, por ejemplo para realizar control de admisión asegurando QoS de extremo a extremo. Como se mencionó antes Ozturk et al. introducen la idea de la red ficticia. La red ficticia es una red con la misma topología que la real pero donde cada flujo llega a cada enlace interior como si los enlaces anteriores en la red no lo hubieran modificado. Ozturk et al. prueban que la región de admisión de la red ficticia está contenida en la región de admisión de la red real. Es decir, que si se utiliza esta simplificación se tendrá la seguridad que los flujos que se acepten van a cumplir los requerimientos de QoS (como función del ratio de pérdida). Lo que podría suceder es que no se aceptaran más conexiones siendo posible hacerlo. En la red ficticia, conociendo qué agregados de flujos atraviesan cada enlace y una estimación del ancho de banda efectivo a la entrada a la red de cada agregado (que define la función velocidad del gran desvío), es posible conocer la probabilidad de desborde de cualquier buffer interior de la red y por tanto el ratio de pérdida de cualquier camino. Si se conoce el ancho de banda efectivo de cada agregado de flujos que atraviesa un enlace, la función de velocidad del desborde del buffer en ese enlace  $k$  se calcula como:

$$I_1^k(x) = \sup_{s>0} \left( sx - s \sum_{X_m \in k} \alpha_{X_m}(s, 1) \right)$$

Este método basado en la red ficticia es fácil de implementar y puede ser utilizado para control de admisión en línea. El problema es cuánto sobreestima la red ficticia la probabilidad de desborde de un buffer con respecto a dicha probabilidad calculada en la red real. Por lo tanto se analizará en este capítulo en qué condiciones el análisis de la red ficticia coincide con el de la red real. También se analizará, cuando estos análisis no coinciden, cuál es la magnitud del error que se comete.

Un caso particular de este problema ha sido estudiado por Casellas [29]. Casellas estudia el caso de una red con dos nodos donde al segundo nodo sólo llega un agregado del tráfico del primer nodo. Los resultados que se demuestran en este capítulo son generales para cualquier red. El resultado de Casellas se puede deducir como un caso particular del obtenido en este capítulo. Los resultados de este capítulo fueron publicados en [16, 17].

## 4.2. Condiciones y cotas del error usando el análisis de la red ficticia

### 4.2.1. Introducción

Se estudiará un enlace interior  $K$  a la red como en el trabajo de Ozturk et al.. Sea  $\mathcal{M}$  el conjunto de tipos de tráfico que acceden a la red y  $\mathcal{M}_i$  el subconjunto de estos tipos de tráfico que pasan por el enlace  $i$ . Se llamará  $X_k^{i,N}(0,t)$ , al trabajo acumulado de  $N$  fuentes de tipo  $i$  en el enlace  $k$  en el intervalo  $(0,t)$ . Se nota  $X^{i,N}(0,t)$  cuando corresponde al tráfico de entrada a la red.

En la red real, la función de velocidad de la probabilidad de desborde del buffer en el enlace  $K$  es:

$$I_K^R = \inf \left\{ \sum_{i \in \mathcal{M}} I_1^{X^i}(x_i) : \sum_{i \in \mathcal{M}} g_i(x) > C_K, x = (x_i)_{i \in \mathcal{M}} \right\} \quad (4.1)$$

En la red ficticia esta función viene dada por:

$$I_K^F = \inf \left\{ \sum_{i \in \mathcal{M}_K} I_1^{X^i}(x_i) : \sum_{i \in \mathcal{M}_K} x_i > C_K, x = (x_i)_{i \in \mathcal{M}_K} \right\} \quad (4.2)$$

Se asumirá que cada tipo de tráfico es un agregado de  $N$  fuentes *i.i.d.* Esta hipótesis tiene como consecuencia que la función de velocidad  $I_1^{X^m}$  es convexa y que  $I_1^{X^i}(\mu_i) = 0$  siendo  $\mu_i = \lim_{N \rightarrow \infty} \mu_i^N$  donde  $\mu_i^N = E(X^{i,N})/N$ . Los problemas de las ecuaciones 4.1 y 4.2, son problemas de optimización con restricciones. El segundo tiene como ventaja que las restricciones son lineales. Las funciones  $I_1^{X^i}$  son continuas, entonces se pueden resolver los siguientes problemas correspondientes a la red real y a la ficticia respectivamente.

$$(P_R) \left\{ \begin{array}{l} \min \sum_{i \in \mathcal{M}} I_1^{X^i}(x_i) \\ \sum_{i \in \mathcal{M}} g_i(x) \geq C_K \end{array} \right. \quad (P_F) \left\{ \begin{array}{l} \min \sum_{i \in \mathcal{M}_K} I_1^{X^i}(x_i) \\ \sum_{i \in \mathcal{M}_K} x_i \geq C_K \end{array} \right.$$

**Definición 4.1.** *Dados dos problemas de optimización*

$$(P_1) \left\{ \begin{array}{l} \min f_1(x) \\ x \in D_1 \end{array} \right. \quad \text{and} \quad (P_2) \left\{ \begin{array}{l} \min f_2(x) \\ x \in D_2 \end{array} \right.$$

$P_2$  se denomina una relajación de  $P_1$  si  $D_1 \subseteq D_2$  y  $f_2(x) \leq f_1(x) \forall x \in D_1$ .

A continuación se enuncia un teorema clásico de optimización que se utilizará para probar los resultados de este capítulo.

**Teorema 4.1.** *Si  $P_2$  es una relajación de  $P_1$  y  $x_2$  es óptimo de  $P_2$  tal que  $x_2 \in D_1$  y  $f_2(x_2) = f_1(x_2)$ , entonces  $x_2$  es óptimo de  $P_1$ .*

*Demostración.*  $f_1(x_2) = f_2(x_2) \leq f_2(x) \leq f_1(x) \forall x \in D_1 \subseteq D_2$ , entonces  $x_2$  es óptimo de  $P_1$  porque minimiza  $f_1$  y pertenece a  $D_1$ .  $\square$

#### 4.2.2. Condición suficiente

**Teorema 4.2.**  *$P_F$  es una relajación de  $P_R$*

*Demostración.* Dado que las funciones  $I_1^{X^i}$  son no negativas, se verifica que  $\sum_{i \in \mathcal{M}_K} I_1^{X^i}(x_i) \leq \sum_{i \in \mathcal{M}} I_1^{X^i}(x_i)$   $\forall x = (x_i)_{i \in \mathcal{M}}$ . Entonces, se debe probar que:

$$\left\{ x : \sum_{i \in \mathcal{M}} g_i(x) \geq C_K \right\} \subseteq \left\{ x : \sum_{i \in \mathcal{M}_K} x_i \geq C_K \right\}$$

Por definición,  $g_i(x) = 0 \quad \forall i \notin \mathcal{M}_K$  y  $g_i(x) \leq x_i \quad \forall i \in \mathcal{M}_K$  entonces

$$\sum_{i \in \mathcal{M}} g_i(x) = \sum_{i \in \mathcal{M}_K} g_i(x) \leq \sum_{i \in \mathcal{M}_K} x_i$$

y por lo tanto  $\sum_{i \in \mathcal{M}_K} g_i(x) \geq C_k$  implica  $\sum_{i \in \mathcal{M}_K} x_i \geq C_K$ .  $\square$

De la proposición anterior se tiene que si un óptimo del problema de la red ficticia ( $P_F$ ) verifica las restricciones del problema de la red real ( $P_R$ ) y la función objetivo toma los mismos valores en ese punto, entonces también es un óptimo del problema de la red real. A continuación se establecen las condiciones de optimalidad del problema  $P_F$ .

Las condiciones de optimalidad KKT (Karush-Khum-Tucker [104]) para el problema de la red ficticia:

$$(P_F) \begin{cases} \min \sum_{i \in \mathcal{M}_K} I_1^{X^i}(x_i) \\ \sum_{i \in \mathcal{M}_K} x_i \geq C_K \end{cases}$$

son las siguientes:

1.  $\nabla \left( \sum_{i \in \mathcal{M}_K} I_1^{X^i}(x_i) + \lambda(C_K - \sum_{i \in \mathcal{M}_K} x_i) \right) = 0$ , donde  $\lambda$  es el multiplicador de Lagrange .

Entonces se cumple que,

$$\frac{\partial I_1^{X^i}}{\partial x_i}(x_i) = \lambda_i \quad \forall i \in \mathcal{M}_K$$

2.  $\lambda \geq 0$ , entonces  $\frac{\partial I_1^{X^i}}{\partial x_i}(x_i) = \lambda_i \geq 0$ . Si  $\lambda = 0$ ,  $x_i = \mu_i \forall i$ . En este caso  $\sum_{i \in \mathcal{M}_K} I_1^{X^i}(\mu_i) = 0$  y no se considera. Entonces se supone que  $\frac{\partial I_1^{X^i}}{\partial x_i}(x_i) > 0$ , lo que implica que  $x_i > \mu_i$ .
3.  $\sum_{i \in \mathcal{M}_K} x_i \geq C_K$ .
4.  $\lambda \left( C_K - \sum_{i \in \mathcal{M}_K} x_i \right) = 0$ . Dado que  $\lambda \neq 0$ , se debe verificar que  $C_K - \sum_{i \in \mathcal{M}_K} x_i = 0$

Por lo tanto, el óptimo de la red ficticia  $P_F \tilde{x} = (\tilde{x}_i)_{i \in \mathcal{M}_K}$  verifica:

$$\begin{cases} \tilde{x}_i > \mu_i & \forall i \in \mathcal{M}_K \\ \sum_{i \in \mathcal{M}_K} \tilde{x}_i = C_K \end{cases}$$

El siguiente teorema da condiciones sobre la red que garantizan la igualdad de la función velocidad de la probabilidad de desborde del buffer del enlace  $K$  en la red real y en la red ficticia ( $E = I_K^R - I_K^F = 0$ ). Teniendo presente que la red es “feed forward”, es posible establecer una relación de orden entre los enlaces. Se dice que el enlace  $i$  es “previo a” o “menor que” el enlace  $j$  si para un camino, el enlace  $i$  se encuentra antes que el enlace  $j$  en la dirección del flujo.

**Teorema 4.3.** Si  $\tilde{x} = (\tilde{x}_i)_{i \in \mathcal{M}_K}$  es óptimo para el problema de la red ficticia  $P_F$ , y la siguiente condición se verifica para todos los enlaces  $i$  menores que  $K$

$$C_K - \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \mu_j \leq C_i - \sum_{j \in \mathcal{M}_i \setminus \mathcal{M}_K} \mu_j \quad (4.3)$$

entonces  $x^*$  definido por:

$$(x^*)_i = \begin{cases} \tilde{x}_i & \text{si } i \in \mathcal{M}_K \\ \mu_i & \text{si } i \notin \mathcal{M}_K \end{cases}$$

con  $\mu_i = \lim_{N \rightarrow \infty} \frac{E(X_0^{i,N})}{N}$ , es óptimo para  $P_R$ .

*Demostración.* Las funciones objetivo de los problemas de optimización 4.1 y 4.2 toman los mismos valores en  $x^*$  porque:

$$\sum_{i \in \mathcal{M}} I_1^{X^i}(x_i^*) = \sum_{i \in \mathcal{M}_K} I_1^{X^i}(\tilde{x}_i) + \sum_{i \in \mathcal{M} \setminus \mathcal{M}_K} I_1^{X^i}(\mu_i) = \sum_{i \in \mathcal{M}_K} I_1^{X^i}(\tilde{x}_i)$$

Con las consideraciones hechas antes, es suficiente probar que  $x^*$  satisface las restricciones del problema real  $\sum_{i \in \mathcal{M}} g_i(x_i^*) > C_K$ . La función  $g_i$  puede ser escrita como una composición de funciones como en 3.25, entonces si  $\sum_{j \in \mathcal{M}_i} x_j^* \leq C_i \forall i$ , se cumple que  $g_i(x^*) = (x^*)_i$  y

$$\sum_{i \in \mathcal{M}} g_i(x^*) = \sum_{i \in \mathcal{M}_K} (x^*)_i = \sum_{i \in \mathcal{M}_K} \tilde{x}_i = C_K$$

lo que prueba el teorema.

Por lo tanto es suficiente asegurar que:

$$\sum_{j \in \mathcal{M}_i} x_j^* \leq C_i \quad \forall i < K$$

Separando la suma, se obtiene

$$\sum_{j \in \mathcal{M}_i} x_j^* = \sum_{j \in \mathcal{M}_i \cap \mathcal{M}_K} \tilde{x}_j + \sum_{j \in \mathcal{M}_i \setminus \mathcal{M}_K} \mu_j \leq C_i \quad \forall i < K$$

y entonces alcanza con asegurar que

$$\sum_{j \in \mathcal{M}_i \cap \mathcal{M}_K} \tilde{x}_j \leq C_i - \sum_{j \in \mathcal{M}_i \setminus \mathcal{M}_K} \mu_j \quad \forall i < K$$

Dado que  $\tilde{x}$  es óptimo de  $P_F$ , satisface  $C_K = \sum_{j \in \mathcal{M}_K} \tilde{x}_j$ , y por lo tanto

$$\sum_{j \in \mathcal{M}_K \cap \mathcal{M}_i} \tilde{x}_j + \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \tilde{x}_j = C_K$$

Sustituyendo en la ecuación anterior y utilizando la hipótesis, se tiene que:

$$\sum_{j \in \mathcal{M}_K \cap \mathcal{M}_i} \tilde{x}_j \leq C_K - \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \tilde{x}_j \leq C_i - \sum_{j \in \mathcal{M}_i \setminus \mathcal{M}_K} \mu_j \quad \forall i < K$$

lo cual prueba el teorema. □

### Ejemplo

Se considera la red de la figura 4.1 con 3 nodos  $i < j < k$  con capacidades  $C_i$ ,  $C_j$  y  $C_k$  respectivamente. Se considera un tráfico de tipo 1, atravesando todos los nodos, un tráfico de tipo 2 que sólo pasa por el enlace  $i$ , un tráfico de tipo 3 que sólo pasa por el enlace  $j$  y un tráfico de tipo 4 que sólo pasa por el enlace  $k$ . Si se verifica la condición 4.3 para el nodo  $k$ , se sabe que no hay error al analizar la red ficticia en lugar de la red real. La condición resulta ser:

$$\begin{cases} C_k - \mu_4 \leq C_i - \mu_2 \\ C_k - \mu_4 \leq C_j - \mu_3 \end{cases}$$



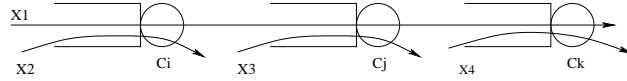


Figura 4.1:

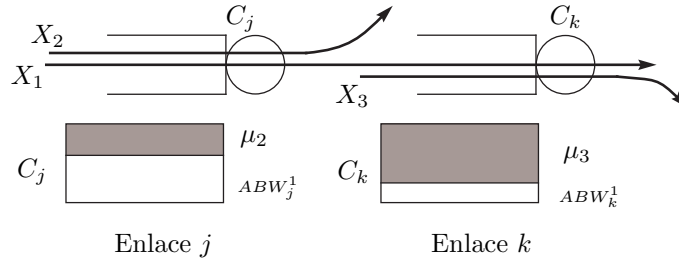


Figura 4.2:

### 4.2.3. Condición suficiente en función del ancho de banda disponible

**Definición 4.2.** Para el tráfico de tipo  $m$  en el enlace  $j$ , el ancho de banda disponible  $ABW_j^m$  se define como la diferencia entre la capacidad del enlace  $j$  y el valor medio de la tasa de transmisión de los otros tipos de tráfico que utilizan el enlace  $j$ .

Teniendo en cuenta la definición anterior, la condición 4.3 del teorema 4.3, asegura que la función velocidad de la probabilidad de desborde del enlace  $K$  en la red ficticia y en la red real coinciden si para todo enlace  $j < K$ , y para todo tráfico de tipo  $m$  en  $K$ ,  $ABW_j^m > ABW_K^m$ . Esta condición se muestra en la figura 4.2 para una red simple con dos enlaces.

### 4.2.4. Condición suficiente pero no necesaria

La condición 4.3 del teorema 4.3 es suficiente para asegurar que la función velocidad de la probabilidad de desborde del enlace  $K$  calculada usando la red real o la ficticia coinciden, pero esta condición no es necesaria. En efecto, si  $\tilde{x}$  es óptimo del problema de la red ficticia, y  $x^*$  definido por:

$$(x^*)_i = \begin{cases} \tilde{x}_i & \text{si } i \in \mathcal{M}_K \\ \mu_i & \text{si } i \notin \mathcal{M}_K \end{cases}$$

satisface las restricciones del problema real, entonces  $x^*$  es óptimo del problema real. Al igual que en la demostración del teorema, si  $x^*$  verifica la siguiente condición

$$\sum_{j \in \mathcal{M}_i} (x^*)_j \leq C_i \quad \forall i < K$$

entonces cumple las restricciones del problema real y es por tanto un óptimo de dicho problema.

Entonces, aunque no se cumplan las hipótesis del teorema anterior, una vez que se encuentra  $\tilde{x}$  óptimo del problema ficticio es fácil verificar si las funciones coinciden. En el caso que no coinciden interesa encontrar una cota para el error cometido al analizar la red ficticia en lugar de la red real.

#### 4.2.5. Cota para el error

Una forma sencilla de encontrar una cota al error es encontrar un punto  $x$  que verifique las restricciones del problema real ya que en ese caso se tiene que

$$E \leq \sum_{i \in \mathcal{M}} I_1^{X^i}(x_i) - \sum_{i \in \mathcal{M}_K} I_1^{X^i}(\tilde{x}_i)$$

Para que  $x$  verifique las restricciones del problema real basta asegurar que

$$\left\{ \begin{array}{l} \sum_{j \in \mathcal{M}_i} x_j \leq C_i \quad \forall i < K \\ \sum_{j \in \mathcal{M}_K} x_j \geq C_K \end{array} \right.$$

Por tanto hay que resolver este sistema de inecuaciones. Dado que se sabe que el óptimo del problema ficticio está en el borde de la región donde se cumplen las restricciones ( $\sum_{i \in \mathcal{M}_K} \tilde{x}_i = C_K$ ) y que interesa encontrar un punto cercano a él, de modo que la cota del error sea lo más pequeña posible, vamos a resolver el siguiente sistema:

$$\left\{ \begin{array}{l} \sum_{j \in \mathcal{M}_i} x_j \leq C_i \quad \forall i < K \\ \sum_{j \in \mathcal{M}_K} x_j = C_K \end{array} \right.$$

Para los casos de interés ( $\sum_{i=1}^M C_i \geq C_K$ ), se puede verificar que este sistema es siempre compatible. En los otros casos no hay pérdidas en el enlace  $K$ , y por lo tanto el análisis con la red ficticia no tiene sentido.

Entonces, se analizará a continuación un algoritmo que permita encontrar una solución a este sistema.

Se define el punto

$$(x^*)_j = \begin{cases} \tilde{x}_j & \text{si } j \in \mathcal{M}_K \\ 0 & \text{si } j \notin \mathcal{M}_K \end{cases}$$

Si verifica las siguientes condiciones

$$\left\{ \begin{array}{l} \sum_{j \in \mathcal{M}_i} x_j \leq C_i \quad \forall i < K \\ \sum_{j \in \mathcal{M}_K} x_j = C_K \end{array} \right.$$

entonces se tiene un punto que cumple las restricciones del problema real pero en algunos casos no sirve para acotar el error pues  $I_1^{X^j}(0) = \infty$ . Si  $P(X_1^{j,N} \leq 0) \neq 0$ , la función  $I_1^{X^j}(0) < \infty$  y se puede obtener una cota del error. Si se supone que  $x^*$  no cumple el sistema de ecuaciones anterior. Entonces se redefinen (disminuyéndolas) aquellas coordenadas que no cumplan que

$$\sum_{j \in \mathcal{M}_i} x_j \leq C_i$$

de esta forma

$$\sum_{j \in \mathcal{M}_i} x_j = C_i$$

Como se tiene que seguir cumpliendo la segunda ecuación, al disminuir algunas coordenadas otras tendrán que aumentar de forma que la suma total siga siendo  $C_K$ . Dado que el sistema es compatible siempre puedo encontrar una solución siguiendo este método. El método no asegura que la solución hallada sea la que minimiza el valor de la cota del error, sin embargo es un método muy simple de implementar y como se verá luego en los ejemplos numéricos las soluciones halladas dan cotas para el error razonables.

#### 4.2.6. La cota del error en un caso particular

Se considera un caso particular donde se cumplen las siguientes condiciones:

1.  $\mathcal{M}_i \setminus \mathcal{M}_K \neq \emptyset \quad \forall i < K$  esto significa que para todo enlace  $i$  menor que  $K$ , existe algún tipo de tráfico que arriba al enlace  $i$  y que no llega al enlace  $K$ .
2.  $\tilde{x}_i \leq C_i \quad \forall i < K$ , lo que implica que  $C_i - \left( C_K - \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \mu_j \right) \geq 0$

Se considera  $\tilde{x} = (\tilde{x}_i)_{i \in \mathcal{M}_K}$  óptimo de  $(P_F)$  y  $x^*$  definido por:

$$(x^*)_j = \begin{cases} \tilde{x}_j & \text{si } j \in \mathcal{M}_K \\ x_j^* & \text{si } j \notin \mathcal{M}_K \end{cases}$$

Si  $x^*$  satisface las restricciones del problema real, se tiene que  $I_K^R \leq \sum_{i \in \mathcal{M}} I_1^{X^i}((x^*)_i)$  y se obtiene la siguiente cota del error:

$$\begin{aligned}
E &\leq \sum_{i \in \mathcal{M}} I_1^{X^i}((x^*)_i) - \sum_{i \in \mathcal{M}_K} I_1^{X^i}(\tilde{x}_i) \\
&= \sum_{i \in \mathcal{M}_K} I_1^{X^i}(\tilde{x}_i) + \sum_{i \in \mathcal{M} \setminus \mathcal{M}_K} I_1^{X^i}(x^*_i) - \sum_{i \in \mathcal{M}_K} I_1^{X^i}(\tilde{x}_i) \\
&= \sum_{i \in \mathcal{M} \setminus \mathcal{M}_K} I_1^{X^i}(x^*_i)
\end{aligned}$$

Dado que

$$\sum_{j \in \mathcal{M}_K} (x^*)_j = \sum_{j \in \mathcal{M}_K} (\tilde{x})_j = C_K$$

es suficiente ver que  $x^*$  cumple que

$$\sum_{j \in \mathcal{M}_i} (x^*)_j = \sum_{j \in \mathcal{M}_i \cap \mathcal{M}_K} \tilde{x}_j + \sum_{j \in \mathcal{M}_i \setminus \mathcal{M}_K} x^*_j \leq C_i \quad \forall i$$

En este caso particular, es posible definir  $x^*_j$  para  $j \in \mathcal{M}_i \setminus \mathcal{M}_K$  tal que

$$\sum_{j \in \mathcal{M}_i \setminus \mathcal{M}_K} x^*_j \leq C_i - \left( C_K - \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \mu_j \right) \quad \forall i$$

y entonces

$$\sum_{j \in \mathcal{M}_i} (x^*)_j \leq \sum_{j \in \mathcal{M}_i \cap \mathcal{M}_K} \tilde{x}_j + C_i - C_K + \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \mu_j$$

Dado que  $\tilde{x}$  es óptimo de  $(P_F)$ , se tiene que  $\sum_{j \in \mathcal{M}_K} \tilde{x}_j = C_K$  y de donde

$$\sum_{j \in \mathcal{M}_i \cap \mathcal{M}_K} \tilde{x}_j = C_K - \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \tilde{x}_j$$

y sustituyendo resulta que

$$\begin{aligned}
\sum_{j \in \mathcal{M}_i} (x^*)_j &\leq C_K - \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \tilde{x}_j + C_i - C_K + \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} \mu_j \\
&= C_i + \sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} (-\tilde{x}_j + \mu_j) < C_i
\end{aligned}$$

porque  $\tilde{x}_j > \mu_j \quad \forall j \in \mathcal{M}_K$  y entonces

$$\sum_{j \in \mathcal{M}_K \setminus \mathcal{M}_i} (-\tilde{x}_j + \mu_j) < 0$$

$x^*$  verifica las restricciones del problema real y por lo tanto es óptimo de dicho problema.

### 4.3. Ejemplos numéricos

**Ejemplo 4.1.** Se considera una red como en la figura 4.3. Se analiza la probabilidad de desborde del enlace  $k$ , asumiendo que  $C_i > C_k$ .

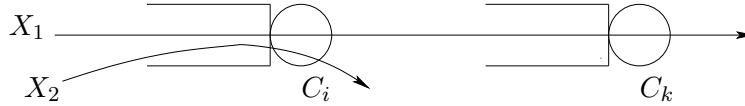


Figura 4.3:

Si se cumple la condición (4.3) para el enlace  $k$ , entonces  $E = I_k^R - I_k^F = 0$ . Esta condición es:

$$C_k \leq C_i - \mu_2$$

Si no se cumple esta condición, como  $\tilde{x} = C_k$  óptimo del problema  $P_F$ , se verificará primero si  $x^* = (C_k, \mu_2)$  es óptimo del problema  $(P_R)$ . Es suficiente mostrar que  $x^*$  verifica las restricciones del problema real, esto es:

$$\begin{cases} C_k + \mu_2 \leq C_i \\ C_k = C_k \end{cases}$$

Si  $C_k + \mu_2 > C_i$ , se busca  $x^* = (x_1^*, x_2^*)$  que verifique

$$\begin{cases} x_1^* + x_2^* \leq C_i \\ x_1^* = C_k \end{cases}$$

Es posible elegir  $x_1^* = C_k$  y  $x_2^* = C_i - C_k > 0$  obteniendo la siguiente cota de error:

$$E \leq I_1(C_k) + I_2(C_i - C_k) - I_1(\tilde{x}_1) = I_2(C_i - C_k) \quad (4.4)$$

En el siguiente ejemplo numérico, se calcula la función velocidad para la red real y para la red ficticia. Sea  $C_i = 16kb/s$  por fuente y  $C_k$  creciendo de 4 to  $15,5kb/s$  por fuente. Todas las fuentes de tráfico son procesos markovianos ON-OFF. Para  $X_1$ , la tasa de bits en el estado ON es  $16kb/s$ , el tiempo medio en el estado ON es  $0,5s$  y  $1,5s$  en el estado OFF. Para  $X_2$ , la tasa de bits en el estado ON es  $16kb/s$ , el tiempo medio en el estado ON es  $1s$  y  $1s$  es el tiempo medio en el estado OFF. Dado que  $\mu_1 = 4kb/s$  la condición de estabilidad es  $C_k > \mu_1 = 4kb/s$ . Usando estos valores, la condición suficiente (4.3) es,  $C_k \leq 8kb/s$ . Las figuras 4.4 y 4.5 muestran que mientras que la condición se verifica, ambas funciones coinciden, pero cuando  $C_k \geq 8kb/s$  se separan. La figura 4.5 también muestra el error de sobrestimación ( $E = I_k^R - I_k^F$ ) y la cota de error (4.4) encontrada antes. En este caso, la cota de error es exactamente el error.

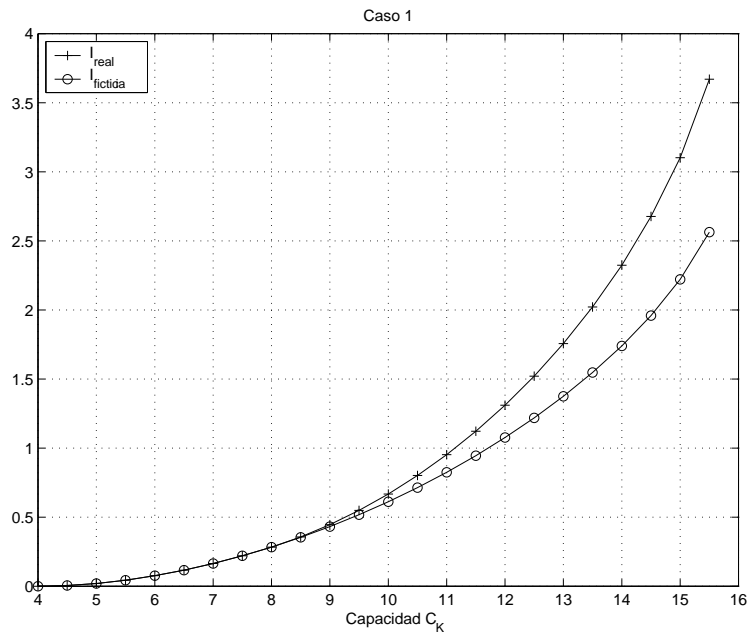


Figura 4.4:

**Ejemplo 4.2.** *Se considera la red de la figura 4.6. Se analiza la probabilidad de desborde del enlace  $k$ .*

Si la condición (4.3) se verifica para el enlace  $k$ , entonces  $E = I_k^R - I_k^F = 0$ . Esta condición es:

$$\begin{cases} C_k - \mu_3 \leq C_i - \mu_2 \\ C_k - \mu_3 \leq C_j \end{cases}$$

Si esta condición no se verifica, y  $\tilde{x} = (\tilde{x}_1, \tilde{x}_3)$  es óptimo de  $P_F$ , primero se verifica si  $x^* = (\tilde{x}_1, \mu_2, \tilde{x}_3)$  es óptimo de  $P_R$ . Es suficiente mostrar que  $x^*$  verifica las restricciones del problema real, esto es:

$$\begin{cases} \tilde{x}_1 + \mu_2 \leq C_i \\ \tilde{x}_1 \leq C_j \\ \tilde{x}_1 + \tilde{x}_3 = C_k \end{cases}$$

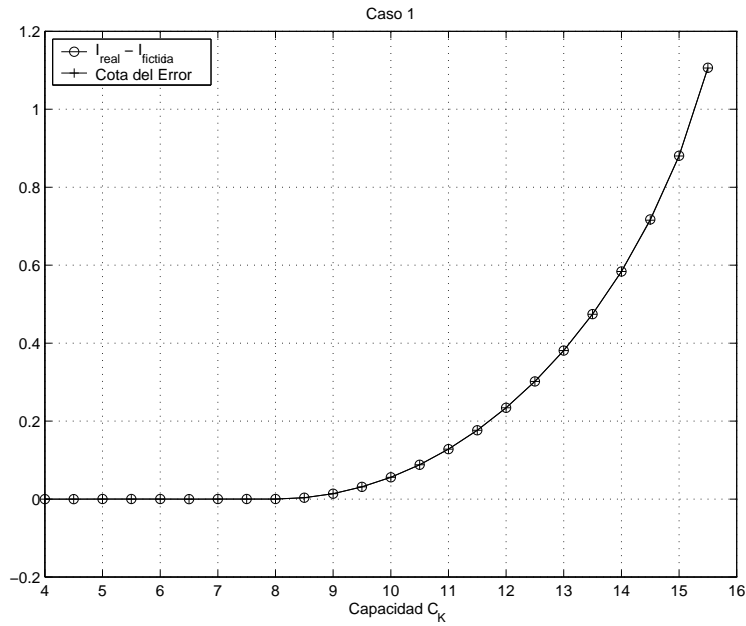


Figura 4.5:

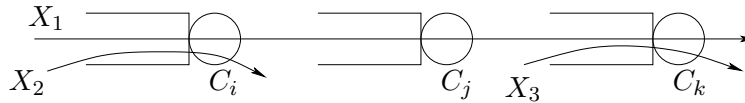


Figura 4.6:

Si estas condiciones no se verifican, se busca  $x^* = (x_1^*, x_2^*, x_3^*)$  que verifique:

$$\begin{cases} x_1^* + x_2^* \leq C_i \\ x_1^* \leq C_j \\ x_1^* + x_3^* = C_k \end{cases}$$

Se elige  $x_1^* = \min(\tilde{x}_1, C_i, C_j)$ . Se identifican tres casos diferentes. En el primer caso  $x_1^* = \tilde{x}_1$ , se elige:

$$\begin{cases} x_1^* = \tilde{x}_1 \\ x_2^* = C_i - \tilde{x}_1 \\ x_3^* = C_k - \tilde{x}_1 \end{cases} \quad (4.5)$$

En este caso la cota de error es:

$$\begin{aligned} E &\leq I_1(\tilde{x}_1) + I_2(C_i - \tilde{x}_1) + I_3(C_k - \tilde{x}_1) - I_1(\tilde{x}_1) - I_3(\tilde{x}_3) \\ &= I_2(C_i - \tilde{x}_1) + I_3(C_k - \tilde{x}_1) - I_3(\tilde{x}_3) \end{aligned} \quad (4.6)$$

Usando que  $\tilde{x}_1 + \tilde{x}_3 = C_k$ , se tiene otra posibilidad para determinar la cota del error. Se puede reescribir la primera ecuación como  $C_k - \tilde{x}_3 + x_2^* \leq C_i$ . Dado que  $\tilde{x}_3 > \mu_3$  se puede elegir  $x_2^* = C_i - (C_k - \mu_3)$  (o cualquier valor menor). En este caso la cota del error es:

$$\begin{aligned} E &\leq I_1(\tilde{x}_1) + I_2(C_i - (C_k - \mu_3)) + I_3(C_k - \tilde{x}_1) - I_1(\tilde{x}_1) - I_3(\tilde{x}_3) \\ &= I_2(C_i - (C_k - \mu_3)) + I_3(C_k - \tilde{x}_1) - I_3(\tilde{x}_3) \end{aligned}$$

La mejor cota de error depende de la posición relativa de los puntos  $C_i - (C_k - \mu_3)$  y  $C_i - \tilde{x}_1$ . Dado que  $C_i - \tilde{x}_1 \leq C_i - (C_k - \mu_3)$ , si ambos son menores que  $\mu_2$  entonces la mejor cota de error es (4.6).

Para el segundo caso  $x_1^* = C_i$  ( $C_i \leq C_j$ ), se elige:

$$\begin{cases} x_1^* = C_i \\ x_2^* = 0 \\ x_3^* = C_k - C_i \end{cases} \quad (4.7)$$

En este caso la cota de error:

$$E \leq I_1(C_i) + I_2(0) + I_3(C_k - C_i) - I_1(\tilde{x}_1) - I_3(\tilde{x}_3)$$

Para el último caso  $x_1^* = C_j$  ( $C_j \leq C_i$ ), se elige:

$$\begin{cases} x_1^* = C_j \\ x_2^* = C_i - C_j \\ x_3^* = C_k - C_j \end{cases}$$

En este caso la cota de error es:

$$E \leq I_1(C_j) + I_2(C_i - C_j) + I_3(C_k - C_j) - I_1(\tilde{x}_1) - I_3(\tilde{x}_3)$$

Para el siguiente ejemplo numérico, se calcula la función velocidad de la probabilidad de desborde para la red ficticia y para la red real. Sea  $C_i = 5,5kb/s$ ,  $C_j = 7kb/s$  por fuente y  $C_k$  variando de 7 to  $25kb/s$  por fuente. Todas las fuentes de tráfico son procesos markovianos ON-OFF. Para  $X_1$ , la tasa de bits en el estado ON es  $8kb/s$ . Para  $X_2$ , la tasa de bits en el estado ON es  $10kb/s$ . Para  $X_3$  la tasa de bits en el estado ON es  $20kb/s$ . El tiempo medio en el estado ON para todas las fuentes de tráfico es  $0,5s$  y para el estado OFF es  $1,5s$ . Dado que  $\mu_1 = 2kb/s$  y  $\mu_3 = 5kb/s$  la condición de estabilidad es  $C_k > \mu_1 + \mu_3 = 7kb/s$ . Usando estos valores, las condiciones suficientes (4.3) son:



$$\begin{cases} C_k \leq C_i - \mu_2 + \mu_3 = 8kb/s \\ C_k \leq C_j + \mu_3 = 12kb/s \end{cases}$$

Estas condiciones se verifican para valores de  $C_k$  menores que  $8kb/s$ . La figura 4.7 muestran que ambas funciones coinciden aun cuando deja de ser válida la condición suficiente y hasta que  $C_k \simeq 15kb/s$ . La razón es que  $x^* = (\tilde{x}_1, \mu_2, \tilde{x}_3)$  es óptimo de  $P_R$ . A partir de ese punto las funciones comienzan a separarse. Las figuras 4.7 y 4.8 también muestran que  $I_k^R$ ,  $I_k^F$  y  $I_k^F + E'$ , donde  $E'$  es la cota de error. Hasta  $C_k = 24kb/s$ ,  $E'$  se calcula usando (4.5), y luego utilizando (4.7). Es importante notar que cuando  $C_k > 14kb/s$  por fuente, la utilización es menor al 50% y por lo tanto, como se puede ver en la figura 4.8 la función velocidad  $I_1^{X^k}$  toma valores mayores a 0.5. Si por ejemplo el número de fuentes que acceden a la red es  $N = 100$ , la probabilidad de desborde es del orden de  $10^{-22}$ . Finalmente se puede ver que la cota de error brinda una buena aproximación y que cuando el error es grande, la probabilidad de desborde es muy pequeña y por lo tanto, estos enlaces no son relevantes para la evaluación de la QoS.

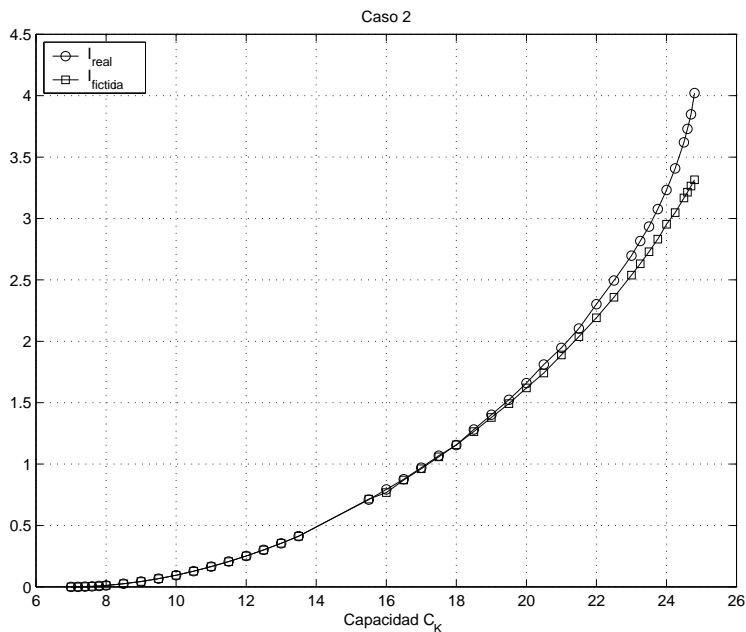


Figura 4.7:

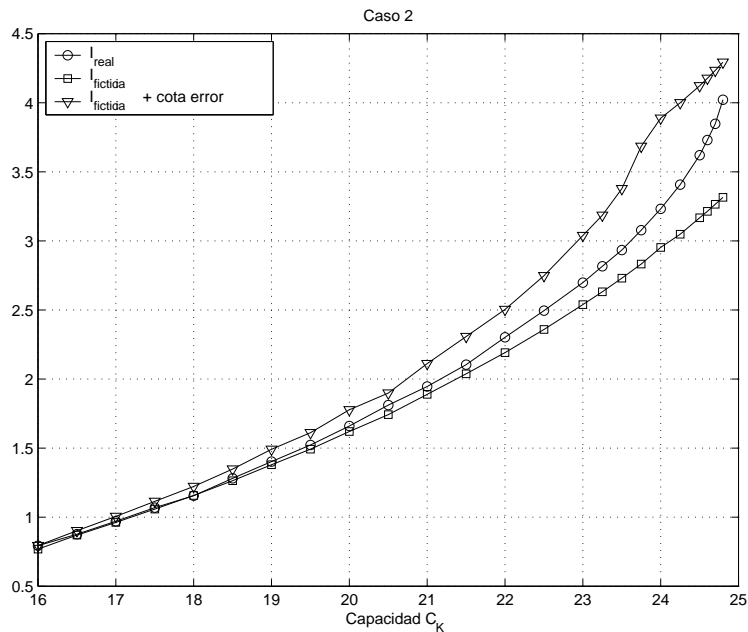


Figura 4.8:

**Ejemplo 4.3.** Se considera la red de la figura 4.9. Se analiza la probabilidad de desborde del enlace  $k$ , asumiendo que  $C_i + C_j > C_k$ .

Si la condición (4.3) se verifica para el enlace  $k$ , entonces  $E = I_k^R - I_k^F = 0$ . Esta condición es:

$$\begin{cases} C_k - \mu_3 \leq C_i - \mu_1 \\ C_k - \mu_2 \leq C_j - \mu_4 \end{cases}$$

Si esta condición no se satisface, como  $\tilde{x} = (\tilde{x}_2, \tilde{x}_3)$  es óptimo de  $P_F$ , se verifica primero si  $x^* = (\mu_1, \tilde{x}_2, \tilde{x}_3, \mu_4)$  es óptimo de  $P_R$ . Es suficiente mostrar que  $x^*$  verifica las restricciones del problema real, esto es:

$$\begin{cases} \tilde{x}_2 + \mu_1 \leq C_i \\ \tilde{x}_3 + \mu_4 \leq C_j \\ \tilde{x}_2 + \tilde{x}_3 = C_k \end{cases} \quad (4.8)$$

Si estas condiciones no se verifican, se busca primero  $x^* = (x_1^*, \tilde{x}_2, \tilde{x}_3, x_4^*)$  que satisfaga

$$\begin{cases} \tilde{x}_2 + x_1^* \leq C_i \\ \tilde{x}_3 + x_4^* \leq C_j \\ \tilde{x}_2 + \tilde{x}_3 = C_k \end{cases}$$

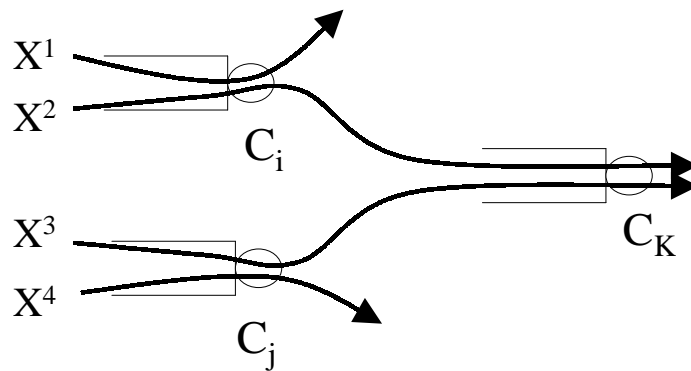


Figura 4.9:

Si  $\tilde{x}_2 > C_i$  or  $\tilde{x}_3 > C_j$  entonces no es posible encontrar tal punto. Por lo tanto, se busca  $x^* = (x_1^*, x_2^*, x_3^*, x_4^*)$  que verifique

$$\begin{cases} x_1^* + x_2^* = C_i \\ x_3^* + x_4^* = C_j \\ x_1^* + x_3^* = C_k \end{cases}$$

Una posible elección es:

$$\begin{cases} x_1^* = C_i \\ x_2^* = 0 \\ x_3^* = C_k - C_i \\ x_4^* = C_j - (C_k - C_i) \end{cases} \quad (4.9)$$

Para el siguiente ejemplo numérico, se calcula la función velocidad para la red real y ficticia. Sea  $C_i = 12kb/s$ ,  $C_j = 14kb/s$  por fuente y  $C_k$  creciendo de 8 a  $25,5kb/s$  por fuente. Todas las fuentes de tráfico son procesos markovianos ON-OFF. Para  $X_1$ , la tasa de bits en el estado ON es  $20kb/s$ . Para  $X_2$ , la tasa de bits en el estado ON es  $16kb/s$ . Para  $X_3$ , la tasa de bits en el estado ON es  $16kb/s$ . Para  $X_4$ , la tasa de bits en el estado ON es  $12kb/s$ . El tiempo medio en el estado ON para todas las fuentes es  $0,5s$  y en el estado OFF es  $1,5s$ . Dado que  $\mu_2 = 4kb/s$  y  $\mu_3 = 4kb/s$  la condición de estabilidad es  $C_k > \mu_1 + \mu_3 = 8kb/s$ . Utilizando estos valores, las condiciones suficientes (4.3) son:

$$\begin{cases} C_k < C_i - \mu_1 + \mu_3 = 11kb/s \\ C_k < C_j - \mu_4 + \mu_2 = 15kb/s \end{cases}$$

Las figuras 4.10 y 4.11 muestran que ambas funciones coinciden aún después de que la condición

deja de cumplirse y hasta que  $C_k \simeq 15kb/s$ . La razón es que  $x^* = (\mu_1, \tilde{x}_2, \tilde{x}_3, \mu_4)$  es óptimo del problema real. A partir de este punto ambas funciones comienzan a separarse.

La figura 4.11 también muestra las funciones velocidad  $I_k^R$ ,  $I_k^F$  y  $I_k^F + E'$ , donde  $E'$  es la cota del error. Hasta que  $C_k = 24kb/s$ ,  $E'$  es calculado usando (4.8), y luego usando (4.9). Al igual que en el ejemplo anterior, cuando  $C_k > 16kb/s$ , la utilización del enlace es menor que el 50% y por lo tanto la cota del error es una buena aproximación para los casos relevantes para la QoS.

Todos los cálculos de la probabilidad de desborde en la *red real* fueron realizados utilizando un paquete de software desarrollado en un proyecto de fin de carrera de Ingeniería Eléctrica. El proyecto, del cual fui el tutor, fue realizado por Buschiazzo, Ferragut y Vázquez [28]. La arquitectura, los algoritmos implementados en el software y ejemplos de cálculo fueron publicados en [27].

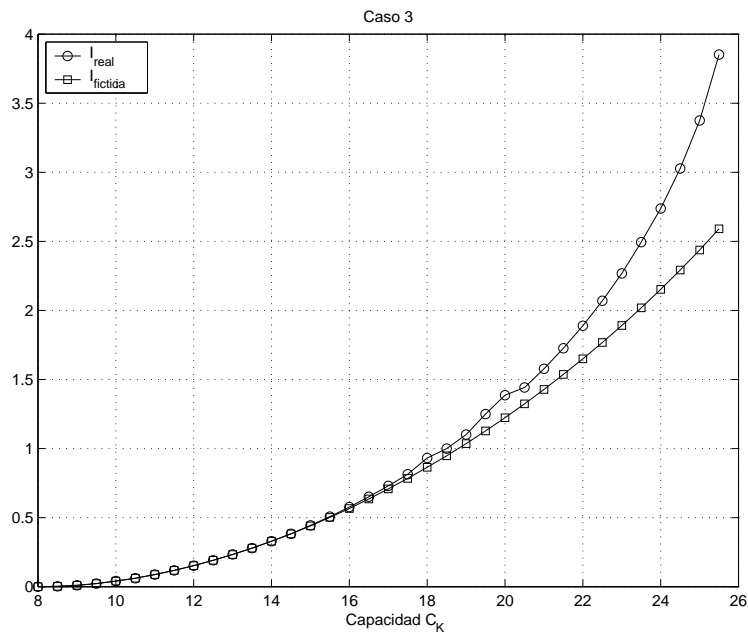


Figura 4.10:

## 4.4. Conclusiones

En este capítulo se analizó que el cálculo de la probabilidad de desborde de un enlace interior y de un camino en la red, es más simple y eficiente utilizando el método de la red ficticia que utilizando la red real. Por este motivo, el análisis de desempeño mediante la red ficticia simplifica (y en algunos casos hace posible) la aplicación del análisis de desempeño en línea por ejemplo para control de admisión.

Generalmente, el análisis con la red ficticia sobre-estima la probabilidad de desborde y la tasa de pérdidas de extremo a extremo. Por lo tanto, esta aproximación si bien es segura, puede sub-utilizar

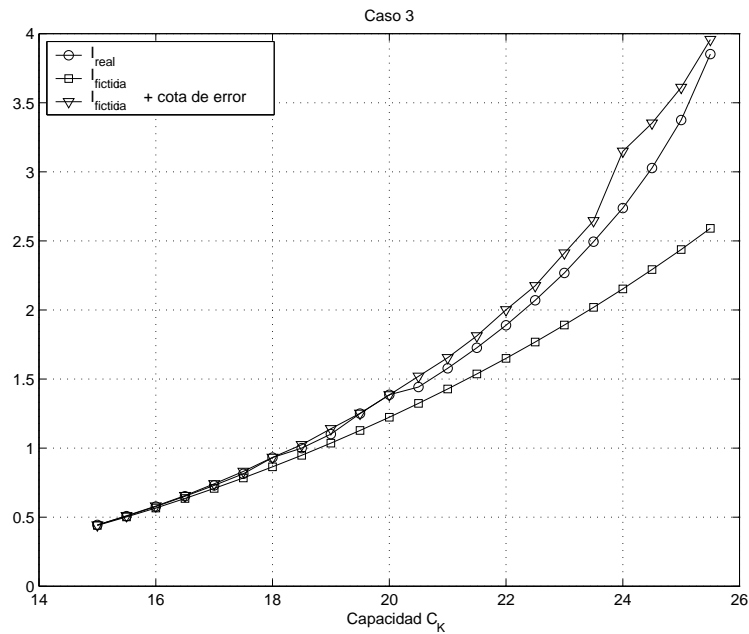


Figura 4.11:

los recursos de la red.

En este capítulo se establece una condición suficiente para que ambos análisis coincidan y que depende sólo de las capacidades de los enlaces y del valor medio del tráfico en la red. Si esta condición se verifica, la probabilidad de desborde calculada utilizando la red ficticia tiene el mismo valor que la probabilidad de desborde calculada utilizando la red real. Cuando esta condición no se verifica, la función velocidad de la probabilidad de desborde calculada utilizando la red ficticia puede ser menor o igual que la la función de velocidad calculada con la red real. Se mostró también en este capítulo que una vez que se calcula la función velocidad de la red ficticia, es muy simple verificar si ambas funciones velocidad coinciden o no. Si no coinciden, se desarrolló un algoritmo simple para encontrar una cota del error.

En los ejemplos numéricos se vio que la cota del error encontrada es ajustada. Se puede ver también, que cuando el error es grande, la probabilidad de desborde es muy pequeña y por lo tanto, esos enlaces no son en general relevantes para calcular los parámetros de QoS.

## Capítulo 5

# Estimación del desempeño y diseño de un enlace

### 5.1. Estimación del ancho de banda efectivo

En los capítulos anteriores se ha visto como calcular la probabilidad de pérdida de un enlace y el ratio de pérdidas en un camino de una red. Para el cálculo de los parámetros de calidad de servicio tanto en el régimen asintótico de muchas fuentes como en el de muchas fuentes y buffer pequeño, es necesario conocer la función logaritmo de la generatriz de momentos del tráfico que arriba a un enlace o a la red. Normalmente por el sentido físico se trabajará con una función del logaritmo de la generatriz de momentos que es el ancho de banda efectivo definido en la ecuación 3.4.1.

En el régimen asintótico de muchas fuentes (figura 5.1) la probabilidad de pérdida viene dada por la optimización de la fórmula *inf sup* (3.17), que repetimos aquí para simplificar la lectura:

$$I(b) = \inf_t \sup_s (s(b + Ct) - \Lambda_t(s\mathbf{1})) \quad (5.1)$$

En el caso de buffer pequeño la diferencia radica en que la escala de tiempo es  $t = 1$  y por lo tanto no es necesario calcular el ínfimo en  $t$ , pero si el supremo de dicha ecuación.

Para resolver dicha ecuación se debe realizar una doble optimización (en el parámetro temporal y en el espacial). De esta doble optimización se obtiene el punto de operación del enlace  $(s^*, t^*)$  ( $(s^*, 1)$  en el caso de buffer pequeño). El problema es que en las aplicaciones en redes operacionales, en general no se cuenta con una fórmula teórica para el ancho de banda efectivo.

En el caso general, se cuenta con trazas de tráfico y se debe resolver la ecuación (5.1), no para  $\alpha(s, t)$  sino para un estimador del ancho de banda efectivo  $\alpha_n(s, t)$ . El primer problema que se estudia en este capítulo es la estimación del ancho de banda efectivo. Posteriormente se considerará la estimación del punto de operación de un enlace y de los parámetros de calidad de servicio.

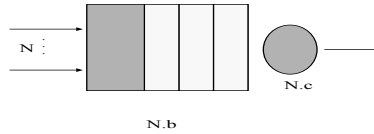


Figura 5.1: Régimen de muchas fuentes

### 5.1.1. Estimación del ancho de banda efectivo

En cuanto a la estimación del ancho de banda efectivo hay dos enfoques, el paramétrico y el no paramétrico. En el enfoque paramétrico se asume un modelo para la fuente de tráfico y a partir de las trazas se estima un conjunto de parámetros de dicho modelo. De esa forma se obtiene un estimador del ancho de banda efectivo y además es posible calcular su intervalo de confianza.

En el enfoque no paramétrico, no se asume un modelo específico del tráfico y se procura construir un estimador del ancho de banda efectivo estimando el valor esperado que aparece en la función generatriz de momentos a través de promedios temporales en la traza.

Este último enfoque, si bien es más general, tiene por desventaja que al no asumir un modelo del tráfico, no se tendrá una expresión analítica del mismo que pueda ser usada para cálculos posteriores. Se analizará primero el enfoque no paramétrico y luego el paramétrico.

### 5.1.2. Estimadores no paramétricos

En este caso el estimador más utilizado para el ancho de banda efectivo utiliza el siguiente procedimiento. Se divide la traza en bloques de largo  $t$  y se construye la siguiente secuencia:

$$\tilde{X}_k = \sum_{i=(k-1)t}^{kt} x(i) \quad 0 \leq k \leq \lfloor T/t \rfloor$$

donde  $x(i)$  es la cantidad de trabajo que arriba por intervalo de tiempo y  $\lfloor c \rfloor$  denota el mayor entero menor o igual que  $c$ .

El ancho de banda efectivo puede entonces estimarse usando el promedio temporal propuesto en [39, 124]:

$$\alpha_n(s, t) = \frac{1}{st} \log \left[ \frac{1}{\lfloor T/t \rfloor} \sum_{j=1}^{\lfloor T/t \rfloor} e^{s\tilde{X}_j} \right] \quad (5.2)$$

A este estimador en la literatura se lo denomina estimador de Dembo.

Es claro en este caso que se tendrá una buena estimación cuando los valores de  $t$  verifiquen que  $t \ll T$ , es decir cuando el número de muestras del proceso de incrementos dentro de la traza sea suficientemente grande.

### 5.1.3. Estimadores paramétricos

El estimador paramétrico más simple asume que la fuente se comporta como un proceso de Poisson y luego estima el parámetro  $\lambda$  de la distribución de Poisson a partir de la media temporal

de la traza. Este modelo, si bien tiene la virtud de la simplicidad, es de poca utilidad práctica ya que en general el tráfico no corresponde a un modelo poissoniano.

Dos casos paramétricos de mayor interés son los llamados modelos de tráfico markovianos. Aquí hay dos modelos: el modelo markoviano continuo y el discreto. Estos dos modelos son similares, la diferencia es que en el primer caso el proceso es gobernado por una cadena de Markov de tiempo continuo mientras que en el segundo es gobernado por una cadena de tiempo discreto. En este modelo se supone que en cada estado de la cadena la fuente transmite a una velocidad (bits/s) constante. Cuando la cadena cambia de un estado a otro se cambia la velocidad de transmisión de la fuente. Un ejemplo simplificado del uso de este modelo es para un codificador ideal de video por diferencia. En este caso una fuente de video codificada se podría modelar de manera simplificada como una cadena con dos estados. Cuando la imagen está “quieta” no hay diferencia entre un cuadro y el anterior y no se transfiere información (estado OFF de la cadena con velocidad de transmisión nula). Cuando la imagen varía mucho se debe transmitir todo el cuadro cada vez y se estaría en el estado ON de la cadena con una velocidad de transmisión alta. En casos reales cuando la imagen está “quieta” en realidad hay poca diferencia y se transmite a baja velocidad (con cierta variabilidad de cuadro a cuadro) y probablemente exista más de un estado de transmisión a velocidad alta dependiendo de la magnitud del cambio.

Se verá a continuación el estimador para el caso continuo y más adelante se verá en un ejemplo el estimador en tiempo discreto.

En el caso del estimador de tiempo continuo, una cadena de Markov de tiempo continuo gobierna el proceso. La cadena tiene  $K$  estados y en cada estado la fuente emite tráfico a una velocidad  $h_i : i \in (1, \dots, K)$ . Si se denomina  $Q$  a la matriz generador infinitesimal de la cadena de Markov,  $\vec{\pi}$  a su distribución invariante, y  $H$  a la matriz diagonal con las velocidades  $h_i$  en la diagonal, el ancho de banda efectivo para una fuente de este tipo es [78, 80] :

$$\alpha(s, t) = \frac{1}{st} \log \left\{ \vec{\pi} \exp [(Q + Hs)t] \vec{\mathbf{1}} \right\} \quad (5.3)$$

donde  $\vec{\mathbf{1}}$  es un vector columna de unos. Este tipo de modelo de tráfico tiene importancia debido a que algunos autores los han propuesto para modelar fuentes de voz o de video.

Perera, Pechiar y Simon [121], encontraron un estimador para el ancho de banda efectivo usando este modelo de tráfico y también una expresión del intervalo de confianza. El cálculo del estimador se basa en estimar la matriz  $Q$ , generador infinitesimal de la cadena, a partir de su estimador de máxima verosimilitud. Este estimador para cada elemento  $q_{ij}$  de la matriz  $Q$  calcula la cantidad de cambios por unidad de tiempo del estado  $i$  al estado  $j$ , contando en la traza la cantidad de pasajes del estado  $i$  al estado  $j$  y dividiendo entre el tiempo que estuvo en el estado  $i$ . A partir de este estimador se estima  $Q_n$  y  $\alpha^n(s, t)$ . Los autores encuentran también una expresión para el intervalo de confianza.

#### 5.1.4. Simulación de los estimadores paramétricos y no paramétricos

Para validar los resultados vistos en la sección anterior se han realizado diversas simulaciones usando tráfico según un modelo markoviano fluido con dos estados (ON-OFF). Cuando la cadena se encuentra en estado ON, el trabajo es producido por la fuente a velocidad constante  $h_0$ , y cuando se encuentra en el estado OFF, no se produce trabajo ( $h_1 = 0$ ).



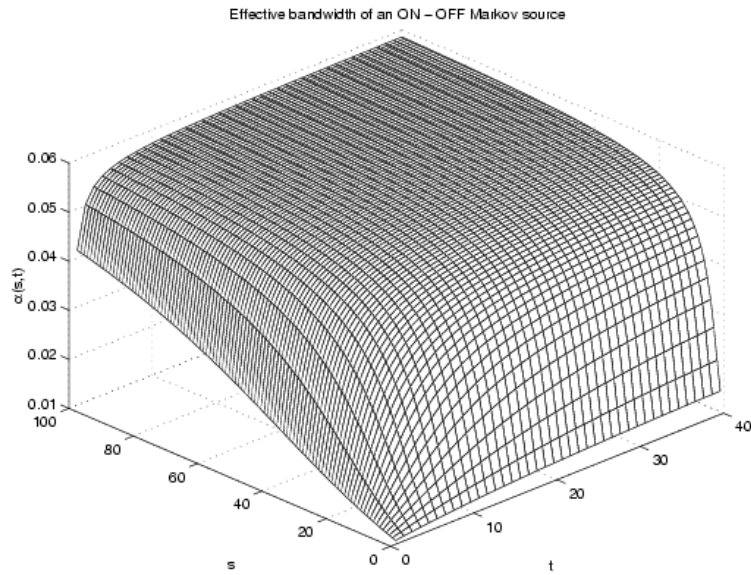


Figura 5.2: Ancho de banda efectivo para una fuente fluida markoviana

En las simulaciones realizadas se generaron trazas de tráfico de largo  $T$  muestras, con la siguiente matriz  $Q$ :

$$Q = \begin{pmatrix} -0,02 & 0,02 \\ 0,1 & -0,1 \end{pmatrix}$$

El ancho de banda efectivo para este proceso teórico calculado según la ecuación (5.3) se muestra en la figura 5.2.

A continuación se compara el ancho de banda efectivo teórico con el obtenido utilizando el estimador paramétrico markoviano mencionado y con el estimador no paramétrico. En la figura 5.3 se muestra el ancho de banda efectivo para  $t = 1$  fijo y en la figura 5.4 para  $t = 50$  fijo. Como se puede apreciar en la figura 5.3 para  $t = 1$  los dos estimadores (paramétrico y no paramétrico) dan una muy buena aproximación del ancho de banda teórico. Sin embargo, para  $t = 50$  el estimador no paramétrico para valores de  $s$  grandes brinda valores inferiores del ancho de banda efectivo teórico. Esto se debe a que para valores grandes de  $t$  se necesita una traza muy larga para tener una buena aproximación. El problema se detecta en valores de  $s$  grande ya que el ancho de banda efectivo cuando  $s \rightarrow \infty$  tiende al valor de pico de la fuente. Por lo tanto, al no ser la traza “lo suficientemente larga”, una sucesión de muchos picos no ocurre y en ese rango de valores se estima mal.

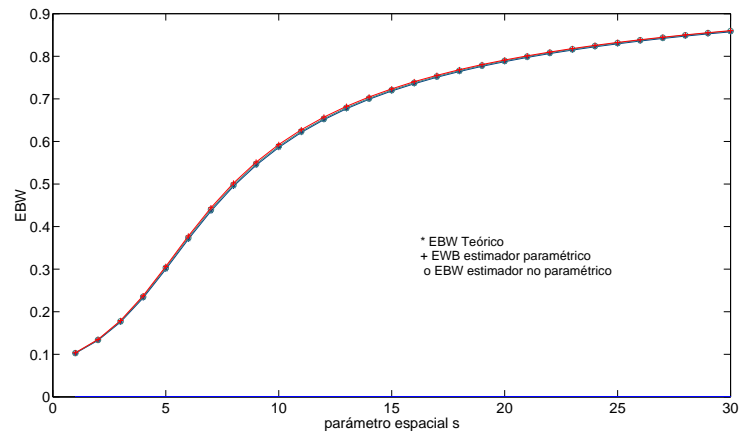


Figura 5.3: Ancho de banda efectivo teórico y estimado para una fuente fluida markoviana para  $t = 1$  fijo

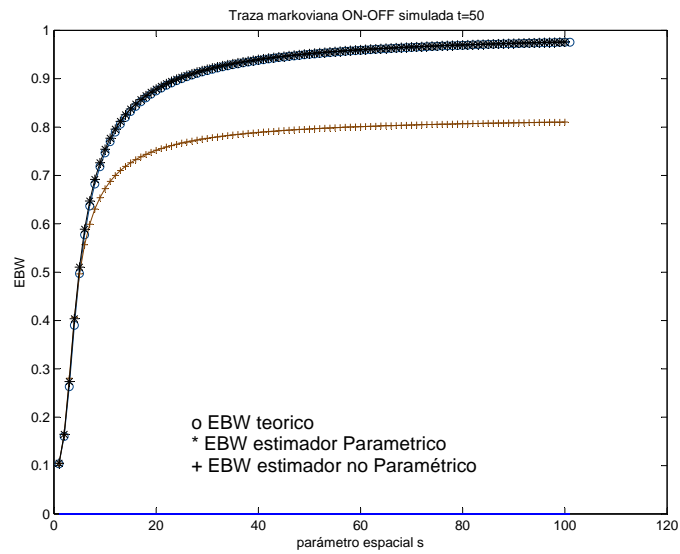


Figura 5.4: Ancho de banda efectivo teórico y estimado para una fuente fluida markoviana para  $t=50$  fijo

### 5.1.5. Dificultades para la aplicación del modelo paramétrico markoviano a trazas reales

En esta sección se verá la aplicación de los estimadores paramétricos markovianos a una traza de tráfico real. Como se verá más adelante este punto presenta diversas dificultades y se propone en este apartado un método para ajustar el estimador paramétrico markoviano.

Para estudiar este problema se analizará su aplicación al cálculo del ancho de banda efectivo de la traza de la figura 5.5, que corresponde a la traza de una fuente de video mpeg. Para este tipo de tráfico distintos autores sugieren modelos markovianos tanto discretos como continuos. Por lo tanto, parece razonable calcular su ancho de banda efectivo utilizando un estimador con alguno de estos modelos.

Para que los cálculos queden más simples, se utilizará el estimador markoviano de tiempo discreto (la misma argumentación valdría para el caso de tiempo continuo). Se analizarán las dificultades para estimar el ancho de banda efectivo con un modelo markoviano discreto. Para simplificar el ejemplo se estimará para un  $t$  fijo, en particular para la menor unidad de tiempo de la traza, es decir, se procurará estimar  $\alpha(s, 1)$ .

Para el caso de tiempo continuo, se vio que el estimador del ancho de banda efectivo se basaba en la estimación de la matriz  $Q$ , generador infinitesimal de la cadena. En este caso el estimador se basará en estimar  $P$ , la matriz de transición de estados. Más precisamente se estimará la ley límite invariante que se llamará  $\pi_\infty$ . Se asume que la cadena es homogénea, regular y de espacio de estados finito y por lo tanto existe una ley límite invariante. Bajo estas hipótesis, se puede probar que [43]:

**Teorema 5.1.** *Sea  $X = (X_n)_{n \in \mathbb{N}}$  una cadena de Markov regular con espacio de estados finito  $E$ . Sea  $f : \mathbb{R} \rightarrow \mathbb{R}$  una función medible. Si se define*

$$\mu_\infty(f) = \sum_{x \in E} f(x) \pi_\infty(x),$$

*el valor esperado de  $f$ , respecto de la probabilidad  $\pi_\infty$ , se cumple que*

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n f(x_i) &\xrightarrow{c.s.} E(f(X)) \\ \frac{1}{n} \sum_{i=1}^n f(x_i) &\xrightarrow{c.s.} \mu_\infty(f) \end{aligned}$$

Usando este Teorema se puede calcular operando  $\alpha(s, 1) = \frac{1}{s} \log E \exp(sX)$ , definiendo  $f(x) = \exp(sh_x)$  se obtiene que

$$\alpha(s, 1) = \frac{1}{s} \log \sum_{x \in E} (\pi_\infty(x) \exp(sh_x))$$

Para estimar el ancho de banda efectivo en este modelo conocidas las velocidades de transmisión en cada estado  $h_x$  es suficiente estimar las componentes  $\pi_\infty(x)$ . Un estimador consistente de este parámetro es  $\pi_\infty(x) = n_x/n$ , siendo  $n_x$  la cantidad de puntos de la traza que se asignen al estado  $x$ , y siendo  $n$  el largo total de la traza.

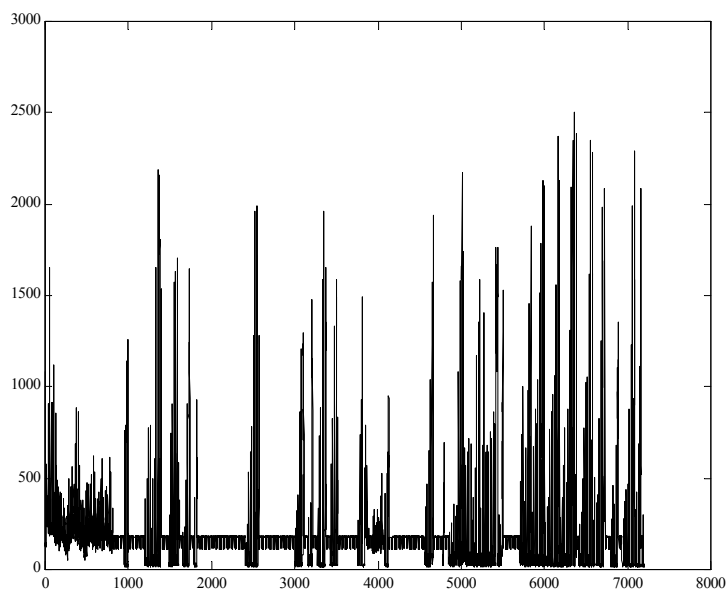


Figura 5.5: Traza de tráfico mpeg (en kbps)

Al tratar de aplicar este estimador en la práctica sobre la traza mpeg surgieron una serie de problemas. En particular para calcular  $n_x$  se debe decidir sobre la traza si cada punto pertenece a un estado o a otro. Para poder tomar esta decisión, se debe dar respuesta a los siguientes problemas:

*¿cuántos son los estados? ¿donde se debe colocar el umbral para decidir que un punto de la traza está en un estado o en otro?*

Por otro lado más arriba se estableció que: “conocidos los  $h_x$ ”, pero *¿cuál es el nivel  $h_x$  de emisión en cada uno de los estados?*

En el caso ideal el nivel es constante pero si se aprecia la traza mpeg en estudio se puede ver que en el caso real hay mucha variabilidad.

Es importante notar que si bien se está trabajando por simplicidad de cálculo con el caso de una cadena de tiempo discreto, estas mismas preguntas surgen para estimar la matriz  $Q$  y la matriz de las tasas  $H$  en el caso de tiempo continuo.

Se analizará el problema por partes. Es obvia la necesidad de definir el mejor modelo en cuanto a cantidad de estados poder realizar una buena estimación. Pero aunque se tuviera una definición por algún procedimiento de la cantidad de estados, el problema de dónde ubicar el umbral para decidir si un punto de la traza pertenece a un estado o a otro, también es relevante. Para visualizar este problema se estimó para  $t = 1$  fijo el ancho de banda efectivo de esta traza, asumiendo un modelo markoviano con dos estados (ON-OFF) y variando el umbral con el cual decidir si un punto de la traza pertenece a un estado o a otro. En la figura 5.6 se muestra que, dependiendo de la elección de umbral, el ancho de banda efectivo estimado puede variar mucho. Se muestra también en el gráfico el estimador no paramétrico de Dembo como curva de referencia.

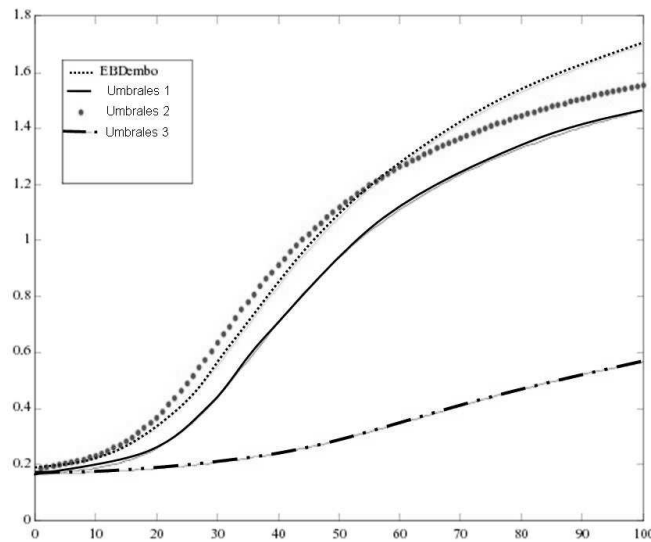


Figura 5.6: EBW con estimador markoviano variando el umbral,  $t$  fijo =1

Para responder a las preguntas anteriores, se propone un algoritmo basado en el siguiente test de markovianidad y en otras consideraciones que se realizarán más adelante. Se asume que el espacio de estados  $E$  tiene  $e$  elementos y que la matriz de transición de estados  $P$  es tal que tiene  $r$  elementos  $p(i, j) > 0$  con  $(i, j) \in E^2$ . Se prueba [43] que

$$Z_n = \sum_{(i,j,k) \in E^3} \frac{(N_{ijk} - N_{ij}N_{jk}/n)^2}{N_{ij}N_{jk}/N_j} \xrightarrow[n]{w} \chi^2(re - r - r + e) \quad (5.4)$$

siendo  $N_{ijk} = \sum_{p=0}^{n-2} \mathbf{1}_{\{X_p=i, X_{p+1}=j, X_{p+2}=k\}}$ , es decir  $N_{ijk}$  es la cantidad de veces que hubo una transición del estado  $i$  al estado  $j$  y luego al estado  $k$ . Análogamente  $N_{ij}$ , es la cantidad de veces que la cadena pasó del estado  $i$  al estado  $j$ .

Lo que dice el teorema anterior es que la variable aleatoria  $Z_n$  converge en distribución cuando  $n \rightarrow \infty$  a una distribución  $\chi^2$ .

Si se asume que todos los términos de la matriz de transición son positivos con el resultado anterior se puede testear la markovianidad de una traza con nivel asintótico  $\alpha$ , usando la región de rechazo  $\{Z_n \geq \chi_\alpha^2(e(e-1)^2)\}$ .

Con base en este resultado se propone el siguiente algoritmo

- Se recorre el número de estados ( $s$ ) desde  $s = 2$  hasta  $s = N$
- Para un número dado de estados  $s$ , se recorren todos los umbrales posibles con un cierto paso dado y se calcula el test de markovianidad anterior para cada caso

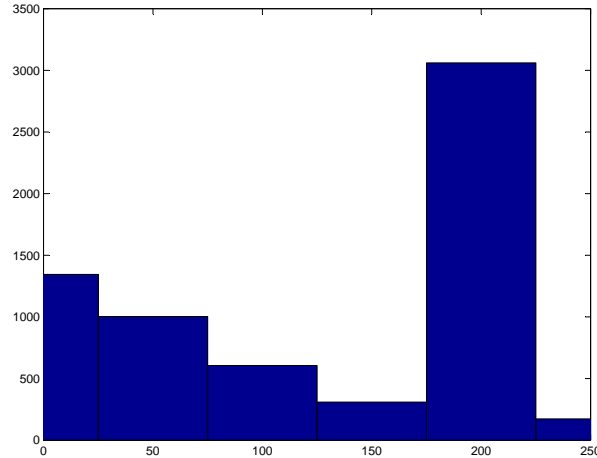


Figura 5.7: Histograma de la traza entre valores de 0 a 250 kb/s

- Se obtiene el conjunto de umbrales para los cuales el error del test sea mínimo.
- Se calcula probabilidad de acierto del modelo para el conjunto de umbrales que con este número de estados obtuvo el mejor resultado.
- Se incrementa el número de estados y se vuelve al primer paso

Finalmente el modelo elegido (cantidad de estados y nivel de umbrales) será aquel que tenga la mayor probabilidad de acierto de todos los estados recorridos.

A la traza mpeg se le aplicó el algoritmo anterior y se obtuvo:

- Para dos estados: Umbral óptimo : 1.79 Mb/s,  $\chi^2 = 0.35$  (dos grados de libertad),  $P(\text{acierto}) = 0,85$
- Para tres estados: Umbrales óptimos: 1.5 y 1.84 Mb/s,  $\chi^2 = 1.55$  (12 grados de libertad),  $P(\text{acierto}) = 0,999$
- Para cuatro estados: Umbrales óptimos 1.65, 1.76 y 1.92 Mb/s,  $\chi^2 = 16$  (36 grados de libertad),  $P(\text{acierto}) = 0,98$

Por lo tanto el modelo elegido fue el de tres estados y con los umbrales antedichos.

Definidos los umbrales, queda aún por definir donde ubicar las velocidades de transmisión  $h_i$ . Para eso se propone analizar el histograma de la traza y ver entre umbrales donde se dan los mayores picos y ubicar en estos valores los  $h_i$ . En la figura 5.7 se muestra el pico del histograma en el rango inferior, por lo que se adoptó  $h_0 = 200kb/s$  y en la figura 5.8 el histograma para los dos rangos superiores, por lo que se adoptó  $h_1 = 1700kb/s$  y  $h_2 = 2100kb/s$ .

En la figura 5.9 se muestra el ancho de banda efectivo según el estimador no paramétrico y el paramétrico usando los estados, umbrales y valores  $h_i$  obtenidos por el algoritmo propuesto. En esta figura se puede apreciar que con el procedimiento propuesto, se logra una buena aproximación.

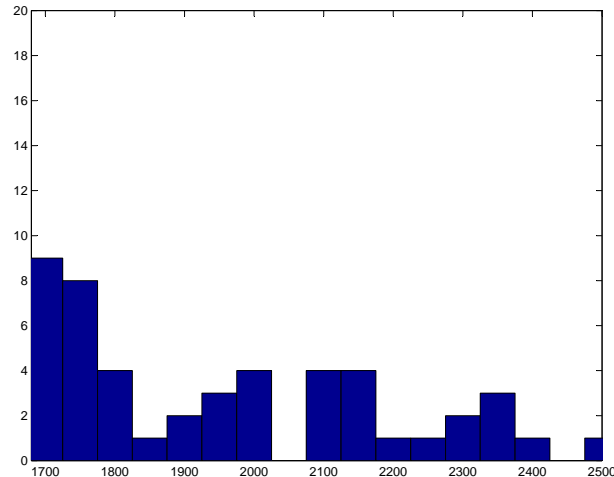


Figura 5.8: Histograma de la traza entre valores de 1700 a 2500 kb/s

Sin embargo, se observó que el resultado es muy sensible a variaciones de los niveles  $h_i$ . Es importante también observar que normalmente el interés no está en tener una buena estimación en todo el rango de valores del parámetro  $s$ , sino solamente en aquellos valores de  $s$  donde está el punto de operación. Teniendo en cuenta esto, puede usarse como complemento al procedimiento anterior el ajustar los  $h_i$  para aproximar la curva del estimador paramétrico al no paramétrico, en la región de interés.

Se puede cuestionar si se tiene un estimador no paramétrico para que se desea ajustar uno paramétrico. La virtud de ajustar un estimador paramétrico es que permite por un lado, conocer un modelo de la fuente de tráfico que luego podrá ser usado para otros cálculos en la red. Además, al conocer el modelo es posible obtener expresiones para el intervalo de confianza del estimador.

## 5.2. Estimación del punto de operación de un enlace

Una vez que se tiene un estimador para el ancho de banda efectivo, surge la pregunta si calculando el punto de operación  $(s_n^*, t_n^*)$  de la ecuación *infsup* con el estimador  $\alpha_n$  se obtendrá un estimador consistente de  $(s^*, t^*)$  y si se podrá probar la normalidad asintótica del estimador del punto de operación. La respuesta de estas preguntas fueron publicadas en el artículo [9]. En dicho trabajo se presenta el siguiente teorema así como las consecuencias y aplicaciones del mismo que se explicarán en lo que resta del capítulo.

**Teorema 5.2.** *Sea  $\alpha(s, t) = \log \Lambda(s, t)$ . Si  $\Lambda_n(s, t)$  es un estimador de  $\Lambda(s, t)$  tal que ambos son*

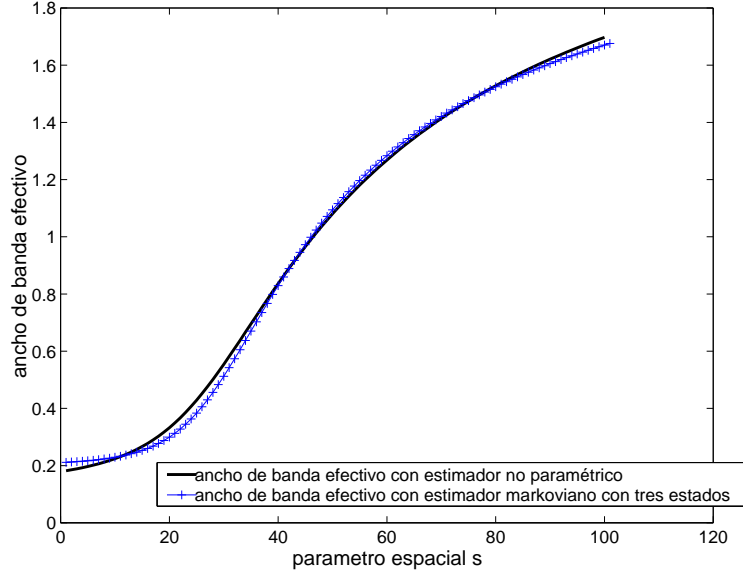


Figura 5.9: Comparación del EBW estimado de la traza

funciones  $C^1$  y:

$$\begin{aligned}\Lambda_n(s, t) &\xrightarrow{n} \Lambda(s, t) \\ \frac{\partial}{\partial s} \Lambda_n(s, t) &\xrightarrow{n} \frac{\partial}{\partial s} \Lambda(s, t) \\ \frac{\partial}{\partial t} \Lambda_n(s, t) &\xrightarrow{n} \frac{\partial}{\partial t} \Lambda(s, t)\end{aligned}$$

casi seguramente y uniformemente sobre intervalos acotados, y si llamamos  $s_n^*$  y  $t_n^*$  a las soluciones de:

$$\begin{aligned}b + ct_n^* - \frac{\frac{\partial}{\partial s} \Lambda_n(s_n^*, t_n^*)}{\Lambda_n(s_n^*, t_n^*)} &= 0 \\ cs_n^* - \frac{\frac{\partial}{\partial t} \Lambda_n(s_n^*, t_n^*)}{\Lambda_n(s_n^*, t_n^*)} &= 0\end{aligned}$$

entonces  $(s_n^*, t_n^*)$  son estimadores consistentes de  $(s^*, t^*)$ . Aún más, si un Teorema Central del Límite funcional (TCL) se aplica a  $\Lambda_n - \Lambda$ , esto es,

$$\sqrt{n} (\Lambda_n(s, t) - \Lambda(s, t)) \xrightarrow[n]{w} G(s, t),$$

donde  $G(s, t)$  es un proceso gaussiano continuo, entonces:

$$\sqrt{n} ((s_n^*, t_n^*) - (s, t)) \xrightarrow[n]{w} N(\vec{0}, \Sigma)$$

donde  $N(\vec{0}, \Sigma)$  es una distribución normal centrada y bivariada matriz de covarianza  $\Sigma$ .



A continuación se brindan las ideas principales de la demostración del teorema. No se detalla aquí la demostración del mismo, si el lector está interesado puede referirse a la publicación antes mencionada [9].

La demostración del teorema se basa en un resultado previo que establece que si una sucesión de estimadores  $\Lambda_n$  converge casi seguramente a  $\Lambda$  con alguna topología adecuada en el espacio  $\mathcal{S}$  de funciones al que pertenece  $\Lambda$  y existe una función  $T : \mathcal{S} \rightarrow \mathbb{R}^2$  que es continua con dicha topología, entonces  $T(\Lambda_n)$  converge casi seguramente a  $T(\Lambda)$ . Si además,  $T$  es diferenciable en  $\Lambda$  y con alguna normalización  $A_n$ , se cumple que  $A_n(\Lambda_n - \Lambda)$  converge a un proceso gaussiano, entonces  $A_n(T(\Lambda_n) - T(\Lambda))$  converge a una normal bivariada. Utilizando este resultado, en la demostración del teorema anterior, primero se observa que  $(s^*, t^*)$  puede escribirse como una función implícita en  $\Lambda$  y de allí se obtiene la función  $(s^*, t^*) = T(\Lambda)$  que permite aplicar el resultado anterior. Definiendo una topología adecuada en el espacio de funciones al que pertenece  $\Lambda$  ( $\mathcal{C}^1$ ) y utilizando las hipótesis del teorema sobre las propiedades de  $\Lambda_n$  se deriva la consistencia y la distribución asintótica del punto de operación a partir de la consistencia y la distribución asintótica del estimador  $\Lambda_n$ .

**Observación 5.1.** *Dado que la convergencia asegurada por el teorema 5.2 es uniforme sobre intervalos acotados, se puede afirmar también que  $\gamma_n$  dado por:*

$$\gamma_n = s_n^*(b + ct_n^*) - \Lambda_n(s_n^*, t_n^*)$$

*hereda las propiedades de los estimadores  $s_n^*$  y  $t_n^*$ . Esto significa que,  $\gamma = F(s^*, t^*, \Lambda)$  donde  $F$  es una función diferenciable. También,  $\gamma_n = F(s_n^*, t_n^*, \Lambda_n)$ . Por lo tanto, si el estimador  $\Lambda_n$  verifica un TCL funcional se obtiene para  $\gamma_n$ :*

$$\sqrt{n}(\gamma_n - \gamma) \xrightarrow{w} N(0, \sigma^2)$$

*Esta observación permitirá construir intervalos de confianza para los estimadores de la probabilidad de desborde.*

**Observación 5.2.** *Como se verá más adelante en este capítulo, existen expresiones para el buffer  $b$  y la capacidad del enlace  $c$  obtenidas por Courcoubetis [40] que son similares a la ecuación inf sup. Por lo tanto, con el razonamiento usado en el teorema anterior se pueden extender los resultados de consistencia y de normalidad asintótica para  $b^*$  y  $c^*$ . Por lo tanto, también se pueden construir intervalos de confianza para estos parámetros de diseño.*

Lo que resta del capítulo se concentra en las aplicaciones del teorema anterior publicadas en el trabajo referido que desde el punto de vista de las telecomunicaciones es quizás lo que presenta más interés.

Este teorema dice que el estimador del punto de operación obtenido a partir de un “buen” estimador del ancho de banda efectivo  $\alpha_n(s, t)$  es consistente y verifica un TCL. Se entiende por “buen” estimador aquel que verifica las hipótesis del teorema anterior. Se puede ver que los estimadores mencionados en este capítulo del ancho de banda efectivo (el no paramétrico y los markovianos) verifican estas hipótesis.

En esta parte de la tesis se analizan los resultados del teorema anterior con trazas de tráfico generadas mediante simulación a partir de un modelo teórico conocido para poder hacer comparaciones. Se usó el modelo de fuente markoviana ON-OFF ya mencionado y como estimador del ancho de banda efectivo se utilizó el estimador no paramétrico visto en la sección anterior.

Para calcular el punto de operación  $(s^*, t^*)$  del modelo teórico de Markov, y su estimador  $(s_n^*, t_n^*)$  para cada traza simulada, se debe resolver el problema de optimización de la fórmula *inf sup* de la ecuación (5.1). En el caso que se está analizando se resolverá dicha optimización con  $\alpha(s, t)$  de la ecuación teórica (5.3) para el modelo de una fuente markoviana y con  $\alpha_n(s, t)$  estimado según el procedimiento visto en el apartado anterior para cada traza.

Sea  $g(s, t) = (Ct + B)s - st\alpha(s, t)$ .

Para resolver la ecuación *infsup*:  $\inf_t \sup_s (g(s, t))$ , se procede numéricamente en dos pasos. En el primero para  $t$  fijo, se encuentra el punto  $s^*(t)$  que maximiza  $g(s, t)$  como función de  $s$ . Se puede demostrar que  $st\alpha(s, t)$  es una función convexa de  $s$ . Esta propiedad de convexidad es usada para resolver el problema de optimización anterior. Su solución se reduce a encontrar la máxima diferencia entre una función lineal con  $s$   $((Ct + B)s)$  y una función convexa por lo que la optimización puede hacerse de manera muy eficiente. Luego de calcular  $s^*(t)$  para cada  $t$ , es necesario minimizar la función  $g(s^*(t), t)$  y encontrar  $t^*$ . Para este segundo problema de optimización no hay propiedades generales que permitan realizar el algoritmo de búsqueda de manera eficiente y una estrategia de búsqueda lineal debe ser usada.

A continuación se muestran resultados numéricos para la estimación del punto de operación y la elipse de confianza del mismo. Para este análisis se simuló un número de trazas  $K = 300$ , de largo  $T = 100000$  muestras y se construyó, para cada traza simulada indexada por  $i = 1, \dots, K$  el correspondiente estimador  $(s_n^*(i), t_n^*(i))$ . Por el teorema 5.2 el vector  $\sqrt{n}((s_n^*, t_n^*) - (s^*, t^*))$  es asintóticamente un vector normal bi-variado con media  $(0, 0)$  y matriz de covarianza  $\Sigma$ . Se estimó la matriz  $\Sigma$  usando las covarianzas empíricas

$$\{\sqrt{n}((s_n^*(i), t_n^*(i)) - (s^*(i), t^*(i)))\}_{i=1, \dots, K}$$

dada por:

$$\Sigma_K = \frac{n}{K} \begin{pmatrix} \sum_{i=1}^K (s_n^*(i) - \bar{s}_n^*)^2 & \sum_{i=1}^K (s_n^*(i) - \bar{s}_n^*) (t_n^*(i) - \bar{t}_n^*) \\ \sum_{i=1}^K (s_n^*(i) - \bar{s}_n^*) (t_n^*(i) - \bar{t}_n^*) & \sum_{i=1}^K (t_n^*(i) - \bar{t}_n^*)^2 \end{pmatrix}$$

donde  $\bar{s}_n^* = \frac{1}{K} \sum_{i=1}^K s_n^*(i)$  y  $\bar{t}_n^* = \frac{1}{K} \sum_{i=1}^K t_n^*(i)$ .

Luego, se puede decir que aproximadamente:

$$(s_n^*, t_n^*) \approx N \left( (s^*, t^*), \frac{1}{n} \Sigma_K \right)$$

de donde un nivel  $\alpha$  de región de confianza se puede obtener como:

$$R_\alpha = (s_n^*, t_n^*) + \frac{A_K^t B \left( \vec{0}, \sqrt{\chi_\alpha^2(2)} \right)}{\sqrt{n}}$$

siendo  $A_K$  la matriz que verifica  $A_K^t A_K = \Sigma_K$  y  $B(x, r)$  la bola de centro  $x$  y radio  $r$ .

Para verificar los resultados se calculó el punto de operación  $(s^*, t^*)$  de otras 300 trazas independientes de las usadas para estimar  $\Sigma_K$ . Se construyó entonces la región de confianza de nivel

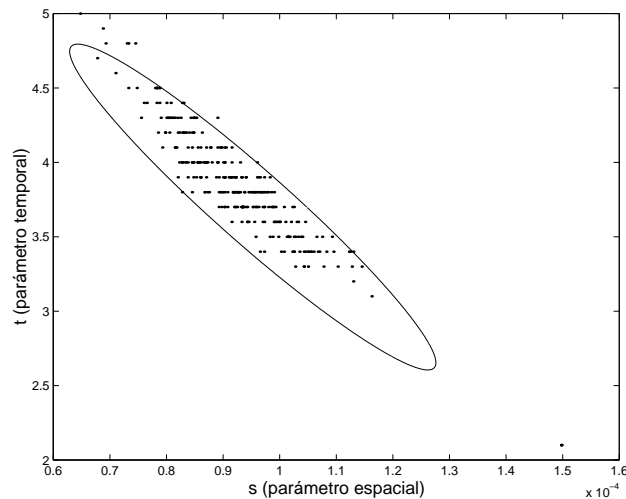


Figura 5.10: Punto de operación estimado y región de confianza con nivel 95 %

95 %. Si los resultados son correctos, aproximadamente el 95 % de los puntos simulados  $(s_n^*, t_n^*)$  deben caer dentro de la región  $R = (s^*, t^*) + \frac{1}{\sqrt{n}} A_K^t B \left( \vec{0}, \sqrt{\chi_{0,05}^2(2)} \right)$ .

Los resultados numéricos se muestran en la figura 5.10, donde se verificó que el 95,33 % de los puntos estimados se ubicaron dentro de la región predicha.

### 5.3. Estimación de los parámetros de calidad de servicio de un enlace

Como se mencionó antes, la estimación del punto de operación tiene como objetivo estimar la probabilidad de pérdida del enlace según la ecuación (5.1) y otros parámetros de calidad de servicio como por ejemplo el retardo. Respecto de este último es importante observar que, en el régimen asintótico de muchas fuentes, el retardo real que sufren los paquetes que atraviesan un enlace coincide asintóticamente con el retardo virtual de los mismos [134]. Por retardo virtual se entiende el que se obtiene a través del tamaño de la cola, es decir que si el enlace envía  $C$  paquetes por unidad de tiempo y la probabilidad de que el tamaño de la cola sea superior a  $B$  es  $q$ , entonces la probabilidad de que el retardo sea superior a  $B/C$  será  $q$ . Por lo tanto en este régimen, si se obtiene un estimador de que el tamaño de la cola sea superior a  $B$ , se tiene directamente un estimador de la distribución del retardo real. Se trabajará en la estimación de la probabilidad de pérdida, ya que el análisis del retardo se deduce de la misma ecuación. Como se mencionó antes si se tiene un estimador del ancho de banda efectivo que verifica las hipótesis del teorema 5.2 entonces el estimador

$$\gamma_n = \inf_t \sup_s ((ct + b)s - st\alpha_n(s, t)) \quad (5.7)$$

es consistente y tiene distribución asintótica gaussiana.

Obtenido este estimador, la probabilidad de pérdida se podrá aproximar por

$$q_n = P(Q_N > B) \approx \exp^{-N\gamma_n} \quad (5.8)$$

siendo  $Q_N$  el tamaño de la cola y  $N$  la cantidad de fuentes del sistema. En la figura 5.11 se muestra la estimación de  $\gamma_n$  para 600 trazas simuladas, su valor teórico y su intervalo de confianza. Los resultados numéricos en este caso muestran que el 94,8% de los valores caen en el intervalo de confianza de 95%.

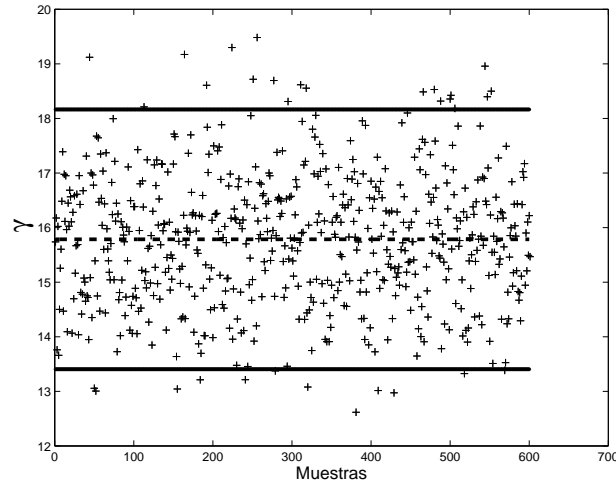


Figura 5.11:  $\gamma_n$  estimado,  $\gamma$  teórico e intervalo de confianza de 95%.

Estos resultados y los de la siguiente sección marcan la importancia del teorema 5.2. A partir de dicho teorema, por un lado, se puede trabajar con estimadores construidos a partir de una traza de tráfico y se sabe que los estimadores son consistentes. Pero además el teorema brinda las bases para construir un intervalo de confianza para los parámetros de desempeño de la red, lo cual desde el punto de vista de la ingeniería es sumamente importante.

## 5.4. Diseño de un enlace basado en la estimación del ancho de banda efectivo

El resultado más interesante para el área de telecomunicaciones es la aplicación del resultado anterior al diseño de un enlace.

Se procura conocer el tamaño de buffer mínimo de un enlace dada su capacidad  $C$ , el tráfico que lo atraviesa y la probabilidad de pérdida máxima que se desea tener (o el retardo máximo deseado). Otra posibilidad es que se tenga la misma información que antes pero se fija el tamaño de buffer que se desea y se quiere calcular la capacidad mínima necesaria del enlace para asegurar

la probabilidad de pérdida requerida. Las respuestas a estos problemas de diseño se obtienen de ecuaciones similares a las de la optimización *infsup* vista antes.

El tamaño de buffer mínimo para asegurar probabilidad de pérdida  $\gamma$  está dada por la ecuación [39]:

$$B_n = \sup_t \inf_s (G_n(s, t))$$

$$G_n(s, t) = \frac{(Nst\alpha_n(s, t) + N\gamma)}{s} - Ct$$

y la capacidad mínima necesaria para asegurar probabilidad de pérdida  $\gamma$  es:

$$C_n = \sup_t \inf_s (K_n(s, t))$$

$$K_n(s, t) = \frac{(Nst\alpha_n(s, t) + N\gamma)}{st} - B/t$$

En las figuras 5.12 y 5.13 se muestran las estimaciones de la capacidad mínima del enlace y del tamaño de buffer mínimo. Estos parámetros han sido estimados a partir de las ecuaciones anteriores, para cada una de las 600 trazas simuladas. En las figuras referidas se muestra además el valor teórico del parámetro y el intervalo de confianza.

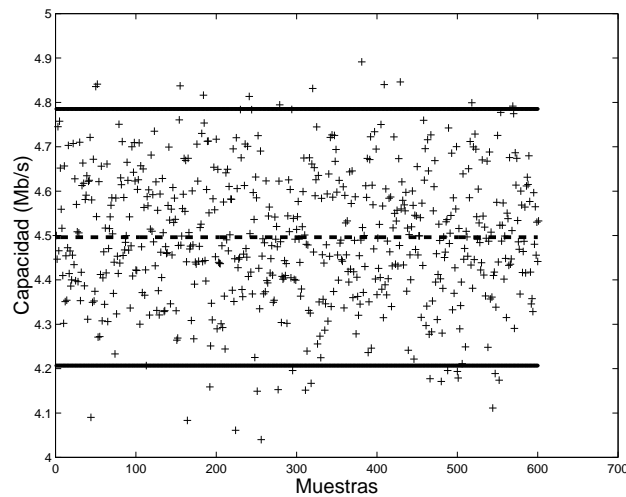


Figura 5.12: Capacidad estimada, capacidad teórica e intervalo de confianza de 95 %.

Los resultados numéricos muestran para el caso de la capacidad que de las 600 simulaciones el 95 % exactamente se ubicó dentro del intervalo de confianza. Los valores negativos del tamaño del buffer para alguna traza indican simplemente que no se necesitó buffer para cumplir con los requerimientos de QoS establecidos.

Como se puede apreciar la variación del tamaño del buffer  $B$  es grande. Este hecho se relaciona con el punto de funcionamiento en el que se encuentra el enlace bajo diseño. En la figura 5.14 se

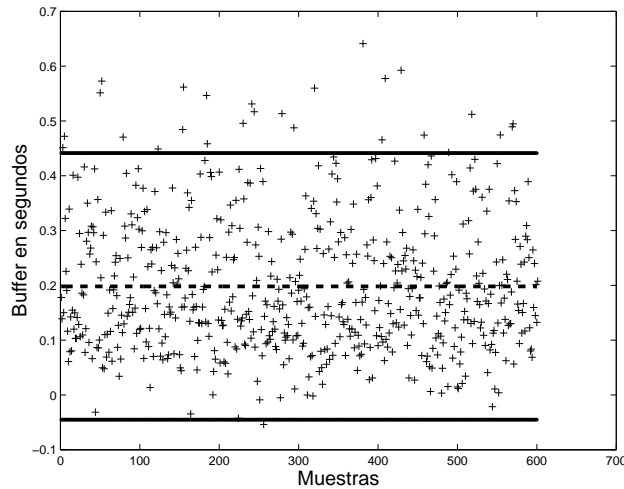


Figura 5.13: Tamaño de buffer estimado, teórico e intervalo de confianza de 95 %.

analiza la curva de  $-\gamma$ , en función del tamaño de buffer. Esta curva es típica de un enlace y en ella se pueden apreciar dos zonas: una para valores pequeños del buffer, donde pequeños cambios en el buffer generan cambios importantes en las pérdidas, y otra para valores grandes del tamaño de buffer, donde para hacer variar las pérdidas mucho se deben realizar cambios importantes en el tamaño del buffer. Como se indica en la figura, en el caso que se está estudiando el enlace está operando en la segunda zona. Las curvas en trazo lleno y en trazo punteado corresponden a distintos  $\alpha$  (por ejemplo  $\alpha$  teórico y un  $\alpha_n$  estimado). Si se pasa de una a otra manteniendo  $B$  fijo,  $\gamma$  varía muy poco. En cambio, si se mantiene  $\gamma$  fijo,  $B$  varía mucho.

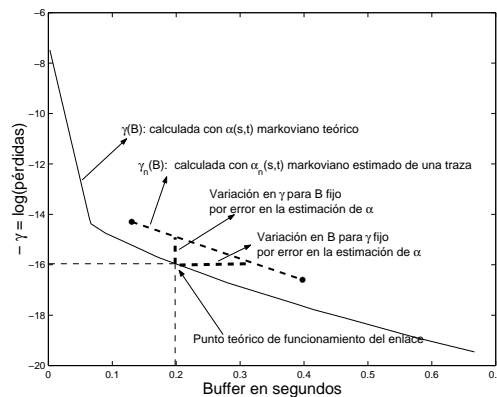


Figura 5.14: Variación de  $-\gamma$  con el tamaño de buffer

De manera similar al caso anterior se puede analizar la variación de  $\gamma$  con la capacidad del enlace  $C$ . En la figura 5.15 se muestra dicha variación. La pendiente de dicha curva crece rápidamente al aumentar  $C$ , y las pérdidas tienden a cero (y por lo tanto  $-\gamma \rightarrow -\infty$ ) al aproximarse la capacidad del enlace al valor de pico de la fuente. En este caso el enlace se encuentra trabajando con una capacidad del 75 % del valor de pico de la fuente. Se observa que si se mantiene  $C$  fijo  $\gamma$  varía poco. A diferencia del caso anterior, si se mantiene  $\gamma$  fijo  $C$  también varía poco. Esto explica la poca dispersión de  $C$  en la figura 5.12.

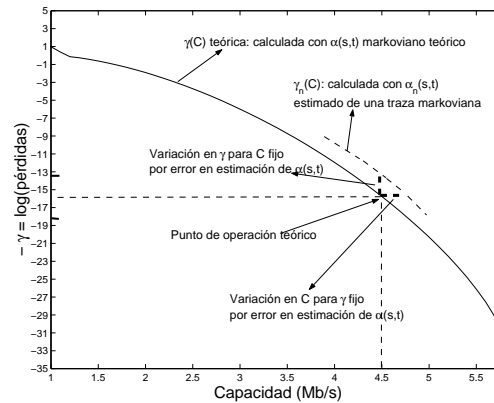


Figura 5.15: Variación de  $-\gamma$  con la capacidad del enlace

## 5.5. Conclusiones

En este capítulo, se estudió como estimar en línea las funciones velocidad de grandes desvíos y se analizaron diversas dificultades existentes para su estimación utilizando diferentes modelos. Un aspecto importante en la estimación de estas funciones es el análisis de la convergencia del estimador del punto de operación. Se encontró un resultado que prueba la consistencia del estimador del punto de operación y también se estableció un TCL para este estimador. Este resultado permite con algunas transformaciones diseñar la capacidad o el tamaño del buffer de un enlace. Se presentó también este método de diseño y que incluye un intervalo de confianza para el diseño realizado.

Por último como resultado de lo estudiado en esta parte de la tesis se observa, que si se tiene un camino en el backbone de la red se pueden utilizar los métodos de este capítulo y el anterior para analizar su desempeño. Sin embargo, hay que señalar que cuando se tiene un camino que pase por la red de acceso y por el backbone, no se tienen resultados que permitan ser aplicados en este caso. En la red de acceso, se puede analizar un enlace con la asintótica de buffer infinito, pero los flujos a la salida del enlace pueden perder la independencia y demás propiedades necesarias para poder aplicar el análisis de desempeño en el siguiente enlace. Por este motivo cuando el camino que interesa analizar es desde un usuario final hasta un servidor o hasta otro usuario final es necesario

complementar los resultados de esta parte con mediciones directas sobre la red. Ese será el objetivo de la siguiente parte.



## Parte II

# Estimación del desempeño en un camino usando mediciones activas

## Capítulo 6

# La estimación de parámetros de extremo a extremo en Internet

### 6.1. Introducción

En una red conmutada de paquetes los servicios sufren diversas degradaciones de su desempeño como por ejemplo: retardo, jitter, y pérdida de paquetes. Se ha generado un reciente desarrollo tanto académico como del IETF (Internet Engineering Task Force) y la ITU (International Telecommunication Union) sobre mediciones en redes IP. El objetivo es establecer un marco, una metodología y una terminología común para medir el desempeño de Internet.

Las diferentes técnicas de medición sobre redes IP y en particular en Internet pueden clasificarse según diversas perspectivas:

- Si la medición es activa o pasiva. Esto quiere decir si se genera tráfico de prueba para hacer las mediciones o si se mide sobre el tráfico que circula normalmente por la red.
- Quién es el actor que realiza la medición. Es diferente el objetivo de un usuario final del objetivo de un operador cuando se realiza una medición.
- Según la capa en que se hace la medición: física, de red, de transporte, de aplicación.
- Según la aplicación de interés. Si se pretende detectar fallas, monitorear un SLA (Service Level Agreement), analizar la calidad de servicio que puede obtener una aplicación, etc.
- La escala de tiempo. Se realizan las mediciones para tomar acciones en el orden de los minutos, segundos o menores, o se las hace para decidir en términos de horas, días o meses.
- Las mediciones se hacen desde un extremo de la red, desde dos extremos de la red o en un equipo o conjunto de equipos de la red.

De acuerdo a la combinación de opciones que se tomen de las diferentes perspectivas, los parámetros a medir y los métodos de medición serán diferentes.

En esta parte de la tesis se analizarán y propondrán diversas técnicas basadas en la medición de diferentes parámetros sobre una red. Se comenzará reseñando en el presente capítulo en la sección 6.2 diversas propuestas existentes en la literatura para estimar dichos parámetros. A partir de esta reseña, se puede observar que la gran mayoría de las técnicas de estimación de extremo a extremo propuestas hasta hace pocos años, han sido desarrolladas para el caso de redes cableadas. Sin embargo, hay particularidades de las redes inalámbricas, y particularmente en redes celulares, que hacen que sea necesario adaptar las técnicas que se encuentran en la literatura. Las redes celulares GPRS/EDGE y 3G/HSDPA se han ido constituyendo en una tecnología importante para el acceso a las redes de datos. Los trabajos en que se analiza mediante mediciones la capacidad de enlaces celulares se concentran en mediciones pasivas sobre un equipo de la red celular [129, 70].

Por este motivo, en el capítulo 7 se analiza la estimación mediante mediciones de extremo a extremo de diversos parámetros relevantes en una red celular.

En los capítulos 8 y 9 se aborda el problema de estimar la calidad de servicio de una aplicación a partir del envío de paquetes de prueba. La propuesta de la tesis se basa en utilizar técnicas de aprendizaje estadístico para este fin.

En el capítulo 8 se resume en primer lugar las técnicas de aprendizaje estadístico que se utilizan en la tesis. Posteriormente, se propone la extensión de una de estas técnicas para el caso no estacionario. Dicha extensión, realizada para el estimador de regresión funcional de Nadaraya-Watson, permite aplicar este estimador al estudio de las redes de datos. Para estimar la calidad de servicio de una aplicación mediante técnicas de aprendizaje estadístico, es necesario estimar mediante medidas el estado de la red. En este capítulo se estima el estado de la red utilizando la distribución empírica de los tiempos entre paquetes de prueba. Utilizando Nadaraya-Watson para el caso funcional no estacionario y dicho estimador del estado de la red, en este capítulo se desarrollan métodos de medición y estimación de la QoS y se aplican a un primer caso de estudio con datos simulados obtenidos con el simulador de redes ns-2.

En el capítulo 9 se analizan las ventajas y desventajas que presenta utilizar otra técnica de aprendizaje estadístico Support Vector Machines (SVM) en lugar de utilizar el estimador de Nadaraya-Watson. Así mismo, se propone un estimador del estado de la red que mejora en varios aspectos a la distribución empírica de los tiempos entre paquetes de prueba que fue el estimador utilizado en el capítulo anterior. Por último, se aplica SVM y este estimador del estado de la red a los mismos datos simulados del capítulo anterior y se comparan resultados.

En el capítulo 10 se explica el software desarrollado para aplicar sobre una red real las técnicas de medición y estimación de la QoS presentadas en la tesis.

En el capítulo 11 se analizan los resultados obtenidos mediante el uso de estas herramientas de software en diferentes redes operacionales.

En el capítulo 12 se enfoca el problema de la calidad de servicio percibida por el usuario (PQoS). Si bien el foco de esta tesis no es la evaluación de la PQoS, esta área es lo suficientemente importante como para realizar un breve análisis de ella, evaluar los resultados existentes y proponer formas de integración de algunos métodos de estimación de la PQoS con las técnicas desarrolladas en la tesis de estimación de la QoS utilizando paquetes de prueba.

## 6.2. Revisión de algunas propuestas en el área de mediciones activas

Existen múltiples trabajos en el área de mediciones activas. En relación con los trabajos de esta parte de la tesis, la mayoría de estos se pueden clasificar en tres grupos principales:

**Estimación de la capacidad de un enlace o de la capacidad del cuello de botella en un camino.** Hay diversos métodos propuestos en la literatura cuyo objetivo es estimar la capacidad de un enlace. Dos ejemplos de interés se pueden ver en [47, 84]. La principal técnica para estimar capacidad se basa en enviar pares de paquetes de prueba de tamaño fijo y analizar en el receptor el tiempo entre arribo de estos paquetes y a partir de esta información estimar la capacidad del enlace. En general todas las técnicas trabajan bien si el tráfico cruzado no es muy importante (tráfico de otros usuarios que se agregan en algún enlace al tráfico de prueba). Sin embargo, cuando se tiene un enlace muy congestionado o en un camino con tráfico cruzado en múltiples enlaces, los errores de estimación pueden ser importantes [84].

**Tomografía de Internet.** El objetivo principal de las técnicas de tomografía de una red es estimar los parámetros de desempeño de un enlace interior desde los extremos de la red [1, 30, 48, 71, 86, 91, 142]. Estas técnicas se basan en enviar tráfico de prueba (en algunos casos utilizando tráfico multicast) y estimar los parámetros internos a partir de diferentes pruebas realizadas desde el exterior de la red.

**“Ancho de banda disponible” de un enlace o un camino.** El ancho de banda disponible (ABW) de un enlace  $i$  en el intervalo de tiempo  $(t, t + \tau)$  se define como

$$A_i(t, t + \tau) = C_i(1 - u_i(t, t + \tau))$$

donde  $C_i$  es la capacidad del enlace y  $u_i(t, t + \tau)$  es la utilización media del enlace en el intervalo de tiempo  $(t, t + \tau)$ . El mínimo  $A_i$  en un camino es definido como el ABW del camino. Hay diversos algoritmos desarrollados para estimar el ancho de banda disponible. Pathload [72, 74, 73], PathChirp [128] y Spruce [133] son ejemplos interesantes de diferentes técnicas de estimación. Strauss et al. [133] comparan Spruce con otras herramientas utilizadas para estimar el ABW, como Pathload. Si bien como se mencionó hay mucha literatura en esta área, para el tipo de aplicaciones que son de interés en esta tesis la estimación mediante medidas del ancho de banda disponible no parece ser un parámetro importante. Este valor medio de capacidad disponible no es fácilmente utilizable para estimar la calidad de servicio que puede sufrir una aplicación específica.

Si bien cualquiera de estas áreas es extensa en cantidad de trabajos producidos, cuando se consideran enlaces celulares se encuentran muy pocos estudios. En particular en cuanto a la estimación de la capacidad de enlaces GPRS/EDGE y su impacto para el análisis de desempeño en tales redes. En el capítulo siguiente se analizan las particularidades de los enlaces celulares y se proponen modificaciones a algunas de las técnicas vistas antes.

## Capítulo 7

# Mediciones de extremo a extremo y su aplicación a enlaces celulares

### 7.1. Introducción

En este capítulo se analiza la estimación de diversos parámetros relevantes en una red celular. En la sección 7.2 se analiza la estimación de capacidad de un enlace GPRS/EDGE y en la sección 7.3 se analiza la estimación del throughput tanto en enlaces GPRS/EDGE como en enlaces 3G/HSDPA. Varios resultados de este capítulo han sido publicados en [18].

### 7.2. Estimación de parámetros en enlaces celulares GPRS/EDGE

#### 7.2.1. Introducción

Los servicios celulares hoy incluyen comunicaciones de voz, de correo electrónico, sms, audio y video streaming, TV móvil, acceso a Internet, etc.. Esta situación entre otras cosas ha sido posible por el constante incremento de la capacidad del acceso a las redes celulares.

Las mediciones de extremo a extremo de redes con acceso *GPRS/EDGE* presentan problemas particulares que no aparecen en el caso de las redes cableadas. El acceso compartido al medio físico, las características del canal de radio frecuencia, así como la prioridad del tráfico de voz sobre el de datos, agregan problemas interesantes a las técnicas existentes.

El foco principal de esta sección es realizar mediciones de extremo a extremo en redes *GPRS/EDGE* a los efectos de estimar la capacidad del enlace de acceso y otros aspectos relevantes en cuanto al desempeño de estas redes. Es de hacer notar que la estimación de la capacidad de un enlace cableado es relevante para una aplicación que ve la red desde fuera (normalmente la aplicación que corre en un equipo del usuario final) y no conoce sus características internas. En el caso de la red celular, estimar la capacidad del enlace de acceso es un parámetro relevante asociado a la QoS también para el operador. La razón fundamental de su utilidad es que la capacidad de un enlace celular *GPRS/EDGE* no es constante (como en el caso cableado) y varía en el tiempo. Las variaciones temporales, dependen de las características del enlace de radio frecuencia y también del grado de

MCS	Tasa de bits por TS (en capa 2, en kbps)	Tipo de modulación
1	8.8	GMSK
2	11.2	GMSK
3	14.8	GMSK
4	17.6	GMSK
5	22.4	8-PSK
6	29.6	8-PSK
7	44.8	8-PSK
8	54.4	8-PSK
9	59.2	8-PSK

Cuadro 7.1: Diferentes esquemas de codificación en EDGE.

congestión de la celda (tanto del tráfico de voz como del tráfico de datos). El análisis de estas variaciones proporciona información relevante de cómo está operando la celda.

Las mayoría de las medidas que se muestran en la parte de resultados en esta sección, fueron realizadas con un software desarrollado para hacer mediciones activas en enlaces celulares. El software fue desarrollado por J. Pereira, J. A. Negreira y S. Pérez como proyecto de fin de carrera bajo mi tutoría [109].

### 7.2.2. Características de las mediciones de desempeño sobre enlaces celulares

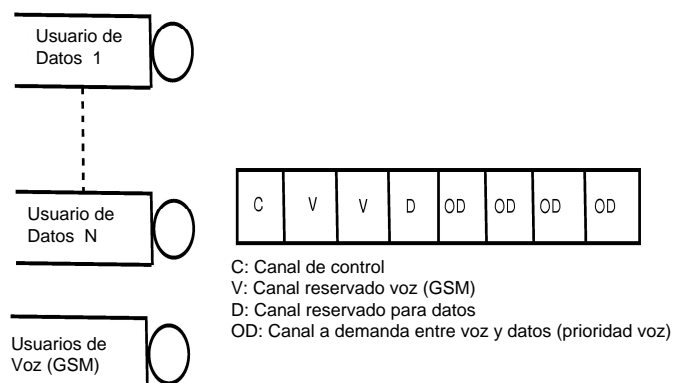
A diferencia del caso cableado, la capacidad en enlaces inalámbricos no es constante. Esto se debe a diversos factores. El común a todos los casos inalámbricos es la relación señal/ruido y por consiguiente la distancia de la radiobase. Cuando la relación señal a ruido es menor la capacidad vista por el usuario es menor, ya que la codificación debe incluir más bits para detectar/corregir errores y por lo tanto la capacidad efectiva de transmisión de datos disminuye. En el caso de GPRS/EDGE, en la cual el canal se multiplexa en el tiempo, esto se traduce en que dependiendo de cada codificación hay una tasa efectiva de transmisión por time slot (TS). En el cuadro 7.1 se muestran las tasas de codificación disponibles en EDGE (MCS).

Otro factor que afecta la capacidad que ve un usuario es la clase del móvil. La clase del móvil, es una clasificación entre los diferentes tipos de terminales que determina cuántos time-slots pueden ser usados por el cliente para transferencia de datos. El cuadro 7.2 lista las clases más comunes y sus características tanto para el enlace de bajada (DL) como para el enlace de subida (UL).

Además en el caso de GSM/GPRS/EDGE hay otros factores que hacen variable la capacidad. Esto se debe a que en esta tecnología, se multiplexa en el tiempo la información tanto de voz como de datos y ambos servicios comparten ciertos conjuntos de slots de tiempo. En general además, las llamadas de voz tienen prioridad respecto a los datos. Por lo tanto, la capacidad que ve un usuario de datos, depende de cuánto tráfico de voz hay en la celda en cierto instante. Además, la cantidad de usuarios de datos hace que a cada conjunto de usuarios de datos se le asignen más o menos slots para compartir en cada grupo. Un usuario comparte con un conjunto de otros usuarios una

Clase	DL slots	UL slots	slots activos
1	1	1	2
2	2	1	3
3	2	2	3
⋮	⋮	⋮	⋮
9	3	2	5
10	4	2	5

Cuadro 7.2: Clasificación de los terminales móviles.

Figura 7.1: Un enlace *GPRS*.

cierta cantidad de slots de datos. La cantidad de slots asignada a cada grupo de usuario de datos, el algoritmo de asignación y el algoritmo que define cómo se comparten los slots dentro de cada conjunto de usuarios, depende del fabricante, pero todos estos efectos influyen sobre la capacidad que ve un usuario.

En las redes GSM/GPRS es posible configurar la cantidad de time slots disponibles para transferencia de datos. Algunas de las variables de configuración en una celda son: el número de time-slots fijos asignados para transferencia de datos, el número de time-slots fijos asignados para telefonía y el número de time-slots “a demanda” compartidos entre telefonía y datos. Como se mencionó, usualmente para estos últimos TS, el tráfico telefónico tiene prioridad preemptiva sobre el tráfico de datos.

En la figura 7.1 se muestra un sistema típico de colas en un enlace *GPRS/EDGE*. Como se puede observar, el sistema incluye interacciones entre el tráfico de voz y el de datos.

Por los motivos anteriores la capacidad en estos enlaces es variable y por lo tanto, en el caso de enlaces celulares GSM/GPRS hablaremos de capacidad instantánea y de capacidad media en un

intervalo. Además, como el usuario no tiene el canal todo el tiempo el canal para su uso exclusivo, se denominará tiempo de actividad del canal al tiempo durante el cual el usuario tuvo el canal para sí.

Del análisis de esta sección se concluye que la capacidad de un enlace celular GSM/GPRS/EDGE no puede ser estimada exactamente con los mismos algoritmos que en el caso de enlaces cableados. Los usuarios celulares no ven una capacidad fija en el tiempo sino que ven una capacidad instantánea que varía en el tiempo. Esta capacidad instantánea será el objeto principal a estimar para luego estimar una capacidad media del enlace vista por un usuario.

### 7.2.3. Metodología de medición

Durante el experimento se enviarán paquetes de prueba de tamaño constante cada tiempo fijo. A los efectos de estimar la capacidad instantánea del enlace cuello de botella, los paquetes de prueba deben ser puestos en la cola juntos en el enlace de acceso celular. La tasa de paquetes para lograr este fin se elige como el máximo de la capacidad del móvil que se está usando para acceder a la red (esto depende de la clase del móvil). Los paquetes se envían durante un corto período a esta tasa.

En el caso de redes cableadas, para cada par de paquetes la capacidad del enlace se estima por  $\frac{\varpi}{\Delta t}$ , donde  $\varpi$  es el tamaño de los paquetes de prueba y  $\Delta t$  es el tiempo entre arribos de paquetes consecutivos. Luego de esta estimación, los valores obtenidos para cada par de paquetes son procesados para filtrar medidas erróneas de la capacidad.

Si un par de paquetes (con tamaño de paquetes  $\varpi$ ) esperan juntos en la cola del enlace cuello de botella (con capacidad del enlace  $C$ ), y no hay ninguna otra cola en el camino, el tiempo entre arribos sería  $\frac{\varpi}{C}$ . Sin embargo, ante la presencia de tráfico cruzado o de otras colas en otros enlaces del camino, las medidas anteriores se ven distorsionadas. Si existe tráfico cruzado entre dos paquetes de prueba, los paquetes se separarán un tiempo mayor que  $T$ . En este caso se estimará una capacidad  $C' < C$ . Otro caso que conduce a situaciones erróneas es cuando los paquetes de prueba deben esperar en una cola posterior a la del cuello de botella que se quiere estimar. En este caso se estimará una capacidad  $C' > C$ . Estos efectos se muestran en la figura 7.2. Por este motivo es necesario un procesamiento posterior que permita filtrar estos casos.

Diferentes técnicas han sido propuestas para filtrar los casos no deseados. Pathrate [47] y Net-timer [84] son dos herramientas conocidas que utilizan diferentes técnicas de estimación de capacidad para enlaces cableados.

La idea principal de estas técnicas, es estimar la función densidad (o el histograma) de los tiempos entre paquetes, y realizar filtrados de los casos espurios. La capacidad estimada será el valor donde se da el máximo de la densidad obtenida. Además de filtrar ciertas situaciones no deseadas, la idea de usar el máximo de la densidad se basa en que los tiempos entre paquetes de prueba tienen una muy fuerte correlación (cuando no hay tráfico entre dos paquetes el valor debería ser siempre el mismo), mientras que las distorsiones por tráfico cruzado o colas que circunstancialmente pueden aparecer en otros enlaces, generan tiempos poco correlacionados y muy dispersos en la función densidad.

Existe un compromiso para la selección del tamaño de los paquetes de prueba. La precisión en el análisis de los intervalos de actividad es importante. A los efectos de tener más muestras en cada intervalo de actividad es necesario enviar paquetes con el menor tamaño posible. Sin embargo, a



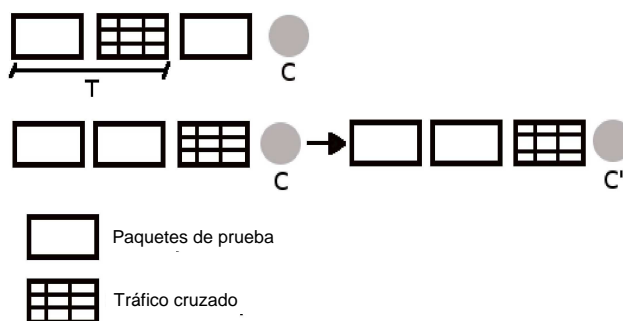


Figura 7.2: Efecto del tráfico cruzado y de encolamiento en un enlace posterior al cuello de botella de un camino.

los efectos de disminuir el error en la estimación de la capacidad es conveniente que los paquetes sean lo más grande posibles.

A los paquetes se les estampa el tiempo de salida y de arribo. El error absoluto de de estas estampas de tiempo efectuados a los paquetes de prueba es del orden de 1 ms. Para resolver este compromiso se realizaron varias pruebas y se encontró que si se utilizan paquetes de 300 – 500 bytes se logra un buen compromiso entre ambos problemas. Además de la fundamentación empírica de la selección de este orden en el tamaño de los paquetes, se debe observar que para paquetes de este tamaño las estampas de tiempo generan un error menor al 10 % para las capacidades más grandes de la tecnología *GPRS/EDGE* que se encuentra en el entorno a los 240 kbps.

#### 7.2.4. Estimación de la capacidad de un enlace celular

Para entender cómo está funcionando una celda *GSM/GPRS* es necesario conocer la capacidad media pero también la capacidad instantánea y la dinámica de variación de la capacidad. De este análisis se puede inferir por ejemplo: horas de saturación del enlace, motivos de dicha saturación (por tráfico telefónico o por tráfico de datos), etc.. Por este motivo, a continuación se propone un algoritmo que extiende los algoritmos de estimación de capacidad para enlaces cableados al caso de un enlace *GSM/GPRS/EDGE*.

Para determinar la capacidad que ve un usuario es necesario en primer lugar determinar los períodos durante los cuales el usuario tiene el canal (períodos de actividad). Luego dentro de cada período será necesario determinar la capacidad que ve el usuario en ese intervalo.

#### Identificación de los períodos de actividad

Para identificar los períodos de actividad de un usuario, se utiliza un algoritmo de aprendizaje no supervisado. En este caso se eligió el método de las *k*-medias por su simplicidad. En este caso el algoritmo calcula un umbral y clasifica el tiempo entre paquetes en 2 clases. El algoritmo clasifica los tiempos entre arribos que corresponden a un mismo período de actividad (grupo 1) y los que corresponden a pasajes de actividad a inactividad (grupo 2). Una vez culminada la clasificación, el umbral se calcula como

$$Th_{Kmeans} = \frac{\bar{g}_1 + \bar{g}_2}{2}$$

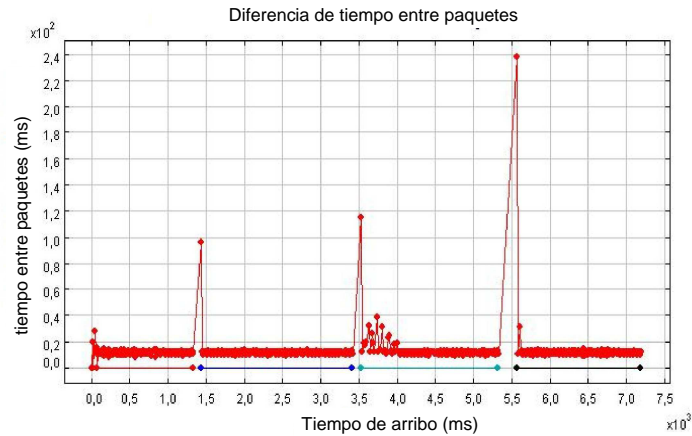


Figura 7.3: Un experimento en el cual los períodos de actividad está claramente diferenciados.

donde  $\bar{g}_1$  y  $\bar{g}_2$  son las medias de los grupos 1 y 2 respectivamente. Cabe mencionar, que puede suceder que por algún fenómeno de ruido tipo impulsivo en el enlace de rf, un paquete se retrase un tiempo importante (por retransmisiones en capa 2). Entre ese paquete y el anterior, se va a detectar un tiempo entre paquetes consecutivos muy elevado (como un pasaje a inactividad). El algoritmo puede detectar este salto entre muestras como un pasaje de un período de actividad a uno de inactividad. Sin embargo, respecto a la estimación de la capacidad que es el objetivo, este problema no afecta.

Un caso particular se da cuando durante un experimento no aparece más que un único período de actividad (cuando hay un único usuario en la celda) y no hay situaciones que generen una modificación sustancial del tiempo de arribo entre paquetes. Para solucionar este problema el algoritmo tiene un umbral mínimo determinado a partir del tamaño de los paquetes de prueba y la capacidad máxima posible del canal,  $Th_{min}$  que si no es superado se asume que hay una única clase y el umbral final viene dado por la ecuación 7.1.

$$Thm = \max \{Th_{min}, Th_{Kmeans}\} \quad (7.1)$$

La figura 7.3 muestra un caso típico de un experimento, en el cual los períodos de actividad están claramente separados. Por otro lado se pueden encontrar casos como en la figura 7.4 en que hay un único período de actividad.

Si no se modificara el algoritmo imponiendo un umbral mínimo, se detectarían múltiples falsos períodos de inactividad.

### Análisis en cada período de actividad

Luego de detectar los períodos de actividad, se analiza cada uno de ellos. En primer lugar se detectan las potenciales capacidades observadas en el período de actividad.

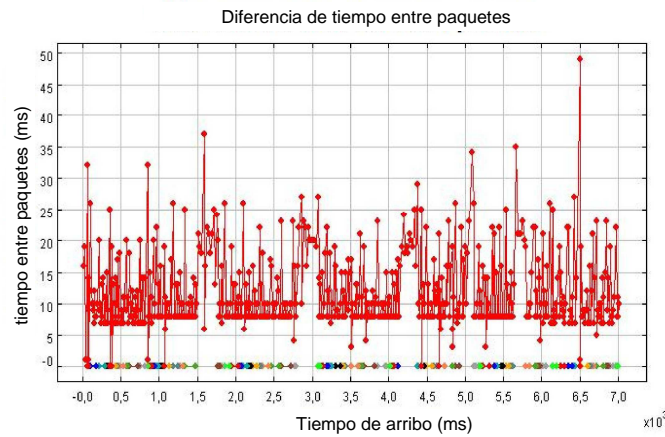


Figura 7.4: Un experimento con un único período de actividad.

#### *Detección de capacidades potenciales*

Para detectar la capacidad observada en cada período de actividad, se realiza una estimación mediante núcleos [132, 21] de la densidad de probabilidad de las potenciales capacidades instantáneas observadas. En este caso para estimar la densidad, se utilizó el Núcleo de Epanechnikov, definido por

$$K(t) = \frac{3}{4}(1 - t^2)1_{\{|t| < 1\}} \quad (7.2)$$

donde

$$1_{\{|t| < 1\}} = \begin{cases} 1 & \text{if } |t| < 1 \\ 0 & \text{en otro caso} \end{cases}$$

Luego de estimada la densidad, el algoritmo estima las potenciales capacidades durante el período de actividad encontrando los máximos de la función densidad.

#### *Procesamiento y filtrado de potenciales capacidades espurias*

Luego de detectadas las capacidades potenciales, se aplican una serie de reglas orientadas a descartar capacidades espurias. Estas pueden darse por diferentes razones: errores de medición, muy malas condiciones de la relación señal a ruido, precisión del sistema de medida, etc. Para esto se aplican reglas que filtran por ejemplo valores capacidad que se sabe no pueden darse en un acceso celular *GSM/GPRS/EDGE*.

#### *Algunos ejemplos en la estimación por núcleos*

La figura 7.5 muestra un caso típico en el análisis de un período de actividad. En este caso la mayoría de las muestras corresponden a la misma capacidad instantánea. Al utilizar la estimación mediante núcleos, otros valores de capacidad que aparecen esporádicamente serán filtrados y se estimará una única capacidad durante el intervalo de actividad.

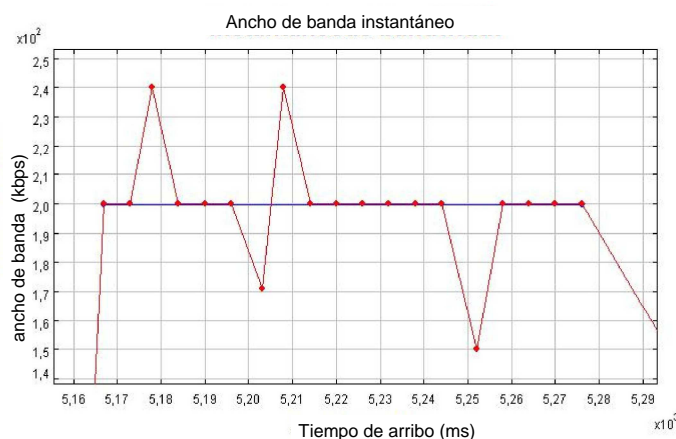


Figura 7.5: Detección de capacidad con bajo nivel de ruido.

Otro caso interesante se puede ver en la figura 7.6. En este experimento la capacidad instantánea oscila entre 120 y 109 kbps. Esta situación es probablemente causada por la precisión de la medida del tiempo entre paquetes. En este caso se utilizaron paquetes de 300 bytes, y el tiempo entre arribos oscila entre 22 y 20 ms. Es razonable suponer que los paquetes arriban al cliente a alguna tasa que corresponde a un tiempo intermedio entre 20 y 22 ms teniendo en cuenta que la precisión temporal es de 1 ms. Esto significa que la capacidad estará entre 109 y 120 kbps. Utilizando la técnica de estimación de densidad por núcleos se determina una capacidad de 113 kbps.

La figura 7.7 muestra un caso en el cual el experimento fue realizado bajo severas condiciones de interferencia. Como se puede ver no es fácil detectar en este caso la capacidad del enlace. En estas condiciones la incertidumbre en la estimación es muy grande por lo que no se reporta estimación alguna en estos casos.

### 7.2.5. Resultados

Se construyó el escenario de pruebas de la figura 7.8. En el mismo se instaló un servidor de mediciones en la red de la Facultad de ingeniería y los móviles se conectaban como clientes a dicho servidor a través de la red celular.

Se realizaron experimentos en diferentes condiciones y ubicaciones. Se consideraron diferentes escenarios variando *el lugar del experimento*, la *relación  $\frac{c}{i}$* , la *hora del día* y el *tipo de móvil*. Los experimentos se realizaron utilizando dos móviles, móvil A y móvil B. Ambos se utilizaron como módem GPRS/EDGE conectados a un PC. Los lugares donde se realizaron las pruebas correspondieron a diferentes zonas de Montevideo y zonas costeras en época de temporada alta.

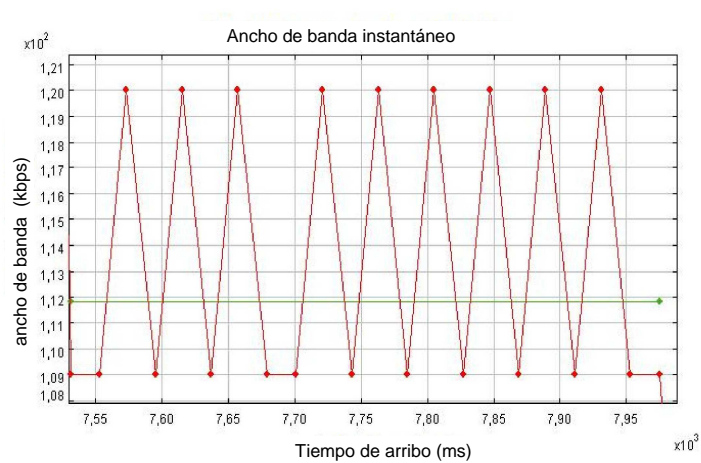


Figura 7.6: Variaciones de capacidad debidas a al precisión del sistema de medición.

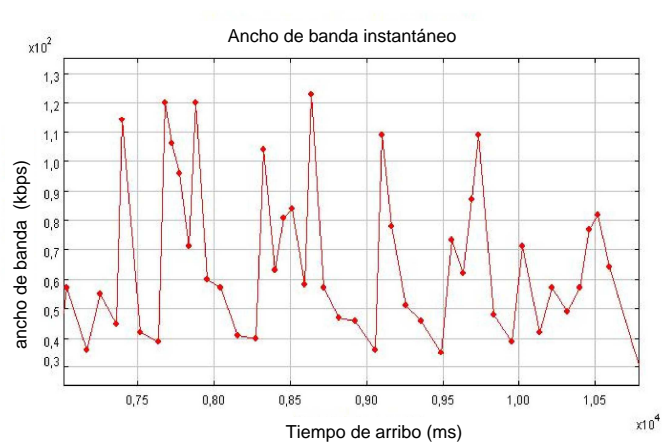


Figura 7.7: Detección de capacidad en condiciones de ruido severo. El algoritmo no es capaz de distinguir una capacidad.

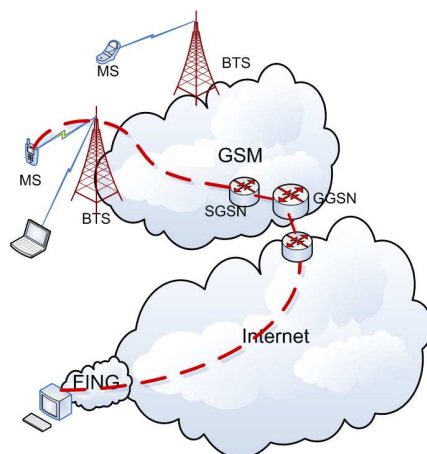


Figura 7.8: Sistema de medición.

### Estimación de la capacidad media

La tabla 7.3 muestra el valor medio, máximo y mínimo registrado en diferentes experimentos en diferentes zonas. En todos los casos se detectó la capacidad media en cada experimento y se reporta el máximo, mínimo y medio de la capacidad media en cada sitio. Todos los experimentos fueron realizados con el Móvil A.

Lugar del experimento	Capacidad Media (kbps)	Capacidad Mínima (kbps)	Capacidad Máxima (kbps)
Montevideo 1	200	167	212
Montevideo 2	202	187	224
Balneario - en la tarde	196	142	216
Balneario- en la noche	142	115	203

Cuadro 7.3: Capacidades medias registradas en diferentes pruebas en diferentes zonas.

En condiciones normales como se puede observar se detectaron capacidades medias que varían aproximadamente entre 190 y 220 kbps. Estos resultados muestran que el móvil A obtiene cuatro TS para bajar con esquemas de codificación entre MCS-6 y MCS-9.

El caso diferente de los vistos en la tabla anterior es el caso de las medidas en un balneario durante la noche. En la figura 7.9 se muestran los diferentes valores detectados en diferentes días y horas. En los primeros tres días se observa que la capacidad media detectada en la tarde es sensiblemente mayor que las capacidades detectadas durante la noche. Esto se debe a que estos experimentos fueron realizados durante un fin de semana largo en un balneario, y en la noche el tráfico de voz es muy importante y reduce la capacidad disponible para datos. También se puede ver que este fenómeno no aparece en el último día (domingo a la noche) ya que una parte importante

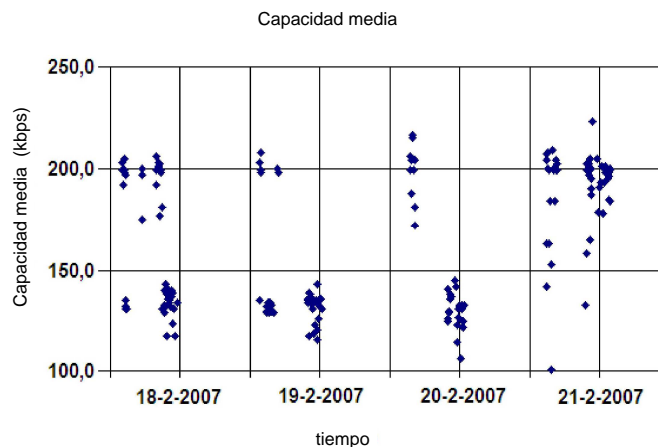


Figura 7.9: Capacidad media en balneario con móvil A.

de los usuarios probablemente estaban retornando.

La figura 7.10 muestra la capacidad obtenida por ambos móviles en una zona de Montevideo. Como puede observarse la capacidad obtenida con el móvil B es menor que la obtenida con el móvil A. Esto se debe a la clase de los móviles.

### 7.2.6. Round Trip Time

Un parámetro importante por ejemplo para aplicaciones TCP es el tiempo de ida y vuelta (RTT) del enlace. Por este motivo se hicieron medidas para evaluar los RTT típicos que se observan en este tipo de enlace. En la tabla 7.4 se muestran los RTT detectados en una celda en Montevideo. En la tabla 7.5 se muestra el RTT para diferentes experimentos en diferentes puntos. Además de la variabilidad del RTT, se observa que los valores medios son grandes comparados con enlaces cableados. En la siguiente sección se estudia el throughput que se puede obtener de un enlace *GPRS/EDGE* y como se verá estos tiempos importantes de RTT tendrán un fuerte impacto.

Móvil utilizado	RTT (ms) Medio	RTT (ms) RTT (Mínimo)	RTT (ms) Máximo
A	1329	833	2318
B	897	709	1803

Cuadro 7.4: RTT registrado en Montevideo 1.

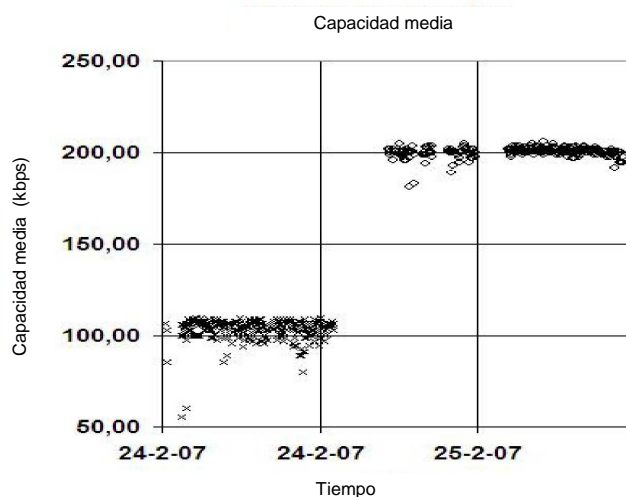


Figura 7.10: Capacidad media con móvil B (cruces) y A (círculos).

Lugar del experimento	RTT (ms) Medio	RTT (ms) Mínimo	RTT (ms) Máximo
Balneario	2376	920	8043
Montevideo 1	1329	833	2318

Cuadro 7.5: RTT obtenido en diferentes zonas con el móvil A.

### 7.3. Throughput vs Capacidad: influencia de diversas características de un enlace celular

Esta sección se concentra en el análisis del throughput que se puede obtener en un enlace celular. El throughput es un parámetro de importancia para las aplicaciones que deben utilizar un acceso celular. Aún si el sistema tiene una importante capacidad física, esta capacidad no puede ser realmente aprovechada por las aplicaciones si el throughput que se puede obtener es pequeño.

El *throughput* de un enlace es un estimador de la tasa de transferencia de datos que una aplicación puede alcanzar mientras que la *capacidad* se refiere a la capacidad “física” real del medio de acceso a la red. Obviamente el máximo throughput teórico que una aplicación puede alcanzar será igual a la capacidad del enlace.

Hay muchos parámetros que afectan el throughput. La capacidad del cuello de botella es un parámetro importante, pero también lo son el protocolo de transporte, el RTT, las pérdidas de paquetes en el camino, etc..

En las secciones siguientes se analiza el throughput en un enlace *GPRS/EDGE* y en un enlace



3G/HSDPA,

### 7.3.1. Estudio throughput/capacidad en un enlace GPRS/EDGE

En la figura 7.11 se muestra la topología utilizada para los experimentos. El cliente utiliza un módem *GPRS/EDGE* mientras que del otro lado hay un servidor, desde el cual se realizan diferentes pruebas.

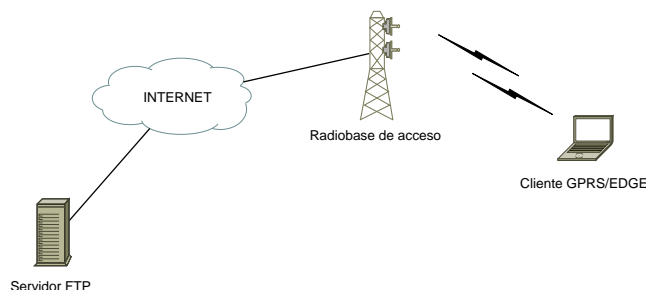


Figura 7.11: Escenario de pruebas GPRS/EDGE.

Como en este caso no interesaba analizar la capacidad del enlace, se realizaron medidas en horarios en que la red se asumía con poco tráfico de voz. Además se verificó que la relación  $\frac{c}{i}$  fuera alta durante todo los experimentos de forma de asegurar que los canales EDGE tuvieran un esquema de codificación (MCS) alto. Con estas hipótesis se asume que el usuario debe obtener tasas altas de transferencia.

#### Análisis del throughput TCP en el enlace GPRS/EDGE de bajada

Este experimento simple consistió tan sólo en bajar varios archivos vía ftp, y analizar el throughput que puede alcanzar una aplicación (figura 7.12).

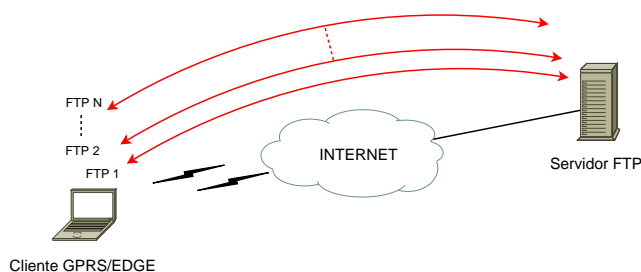


Figura 7.12: Throughput TCP en el enlace de bajada.

En estos experimentos se utilizó una herramienta estándar (Ethereal) para registrar los paquetes que arribaban al PC cliente y los que salían del servidor. Además se utilizó una herramienta para estimar el throughput que arriba al módem (NetStat Live). Se comenzó realizando una bajada de archivo. Luego se fue aumentando el número de transferencias simultáneas de archivos. En cada

caso se registraba el throughput obtenido. Este procedimiento se continuó agregando conexiones simultáneas hasta que el throughput agregado de todas las conexiones dejó de crecer.

### Análisis capacidad vs throughput en el enlace de subida

En este experimento se utilizó la herramienta Iperf, que permite generar tráfico TCP o UDP y obtener estadísticas sobre estas medidas. Se realizaron medidas tanto en el enlace de bajada como en el de subida. Aquí sólo se reproducen como ejemplo las del enlace de subida ya que las conclusiones que se puede extraer en ambos casos son similares.

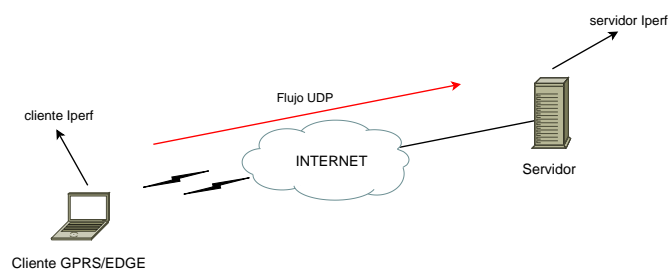


Figura 7.13: Throughput en el enlace de subida.

Los experimentos con esta herramienta dependen del tipo de tráfico utilizado. Cuando se utiliza UDP, la estimación consiste en enviar una o más ráfagas de paquetes UDP desde el cliente al servidor, inundando el enlace de subida y analizando los paquetes que arriban al servidor como se muestra en la figura 7.13. En el caso de tráfico TCP, se abren una o más conexiones TCP, determinando el throughput agregado en el servidor.

### Resultados

La figura 7.14 muestra el throughput alcanzado cuando se fueron abriendo cronológicamente 1, 2, 3, 5, 1, 4, y finalmente 5 conexiones simultáneas.

La tabla 7.6 muestra el throughput medio alcanzado cuando se abren diferente cantidad de conexiones simultáneas.

Estos resultados muestran, que con una conexión TCP en el caso de un enlace GPRS/EDGE se está lejos de obtener la capacidad del enlace. La capacidad del enlace estimada es de aproximadamente 180 kbps, (el móvil está usando 3 slots para bajar (la configuración es 3+2)). Como se muestra el throughput obtenible con una conexión es del orden de la tercera parte de la capacidad. Sin embargo, este valor se incrementa cuando se abren múltiples conexiones simultáneas. Para cinco conexiones simultáneas el throughput agregado es cercano a la capacidad del enlace. Un resultado de este tipo era esperable ya que es sabido que TCP no alcanza la capacidad del enlace debido a los mecanismos de control de congestión que utiliza. Sin embargo, el resultado a primera vista más sorprendente es la magnitud de esta diferencia. TCP sobre enlaces inalámbricos tiene problemas de desempeño porque TCP asume que el medio es confiable y por tanto las pérdidas se deben a congestión. Cuando se detectan pérdidas se reduce la ventana de transmisión de TCP para disminuir la congestión. En un medio poco confiable (como un enlace de radio frecuencia) este

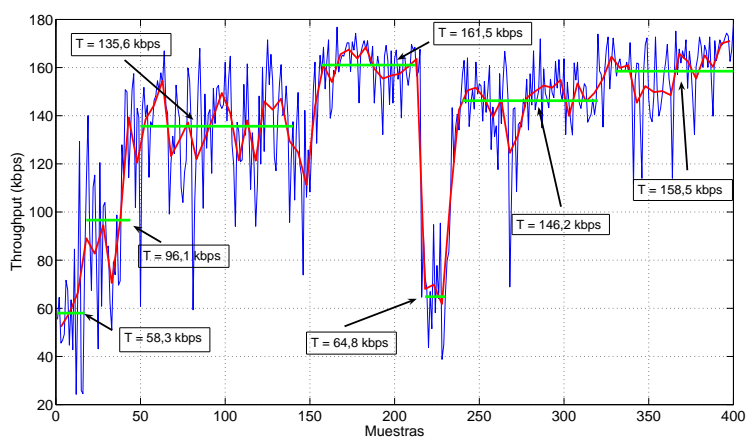


Figura 7.14: Throughput de bajada cuando se abren varias conexiones simultáneas.

Número de conexiones TCP simultáneas	Throughput (kbps)
1	58,3
2	96,1
3	135,6
5	161,5
1	64,8
4	146,2
5	158,5

Cuadro 7.6: Estimación del throughput.

fenómeno tiene consecuencias fuertes sobre el desempeño de TCP, ya que cada vez que por ruido se pierde un paquete TCP, se disminuye la ventana pensando que el problema es de congestión. Para solucionar este problema los enlaces inalámbricos agregan en la capa MAC mecanismos de retransmisión de forma que por encima de la capa 2, el medio sea visto como confiable por protocolos como TCP. Este es el caso de *GPRS/EDGE*, que implementa este tipo de mecanismos. Sin embargo, estos mecanismo que hacen que el medio sea visto como confiable, introducen retardos importantes debido a las retransmisiones. Se sabe también [117], que el throughput de TCP es aproximadamente inversamente proporcional al RTT y a la raíz cuadrada de las pérdidas. Por lo tanto al aumentar el retardo para hacer más confiable el medio, se aumenta el RTT y por tanto se disminuye fuertemente el throughput que se puede obtener de él.

Como se puede ver en la tabla 7.5, en todos los casos el RTT es muy alto comparado con los valores usuales que se observan en un enlace cableado. Obviamente cuanto menos confiable sea el enlace de radio frecuencia y más interferencia tenga peor es el RTT y por tanto peor será el throughput que se puede obtener.

Por otra parte, las tablas 7.7 y 7.8 muestran para el enlace de subida la estimación de capacidad que se obtendría si se usara TCP en el primer caso y UDP en el segundo. Como era esperable por los comentarios anteriores, el throughput TCP no es un buen estimador de la capacidad. Por el contrario, en el caso UDP la estimación de la capacidad es relativamente aproximada (el móvil tenía 2 TS para subir) y no varía por el número de conexiones simultáneas que se abren.

Número de conexiones TCP simultáneas	Estimación (kbps)
1	56,3
2	74,2
3	91,5

Cuadro 7.7: Estimación de la capacidad del enlace de subida con Iperf usando conexiones TCP.

### 7.3.2. Throughput y consideraciones sobre el desempeño de una red 3G/HSDPA.

Los resultados de esta sección fueron realizados en la red celular de un operador y durante el período de instalación de la red 3G. Se aplicaron técnicas de mediciones desde los extremos de la red durante el proceso de implantación de la red. El objetivo era detectar y aportar soluciones a los problemas que fueran surgiendo durante la instalación de la red 3G en particular en HSDPA. El operador había detectado problemas de bajo throughput en las primeras pruebas realizadas durante la instalación, pero no estaban claras las causas ni las posibles vías de solución. Se analizará en lo que sigue de esta sección una técnica que permitió evidenciar el problema y analizar las diferentes causas que estaban afectando el desempeño de la red. Es de hacer notar que el problema analizado para el caso de *GPRS/EDGE* donde el RTT tenía un impacto importante sobre el throughput

Número de conexiones UDP simultáneas	Estimación (kbps)
1	114
2	112
3	117

Cuadro 7.8: Estimación de la capacidad del enlace de subida con Iperf usando UDP.

TCP debería ser menor en este caso. HSDPA tiene mecanismos de detección, corrección de errores y retransmisiones en capa 2 mucho más eficientes. Un RTT típico de un enlace HSDPA está en el orden de 100 ms, que es muy inferior a los valores de RTT obtenidos para GPRS/EDGE como se puede ver en el cuadro 7.5.

En la figura 7.15 se muestra un diagrama general de la topología en la que se desarrollaron los experimentos.

En dicha topología se permite probar varios tipos de conexiones, a saber:

- Conexión entre el equipo de prueba y el servidor `arbolita.fing.edu.uy` vía modem HSDPA (pruebas 3G).
- Conexión entre el equipo de prueba y `arbolita` vía modem EDGE (pruebas 2G).
- Conexión entre el equipo de prueba y `arbolita` vía ADSL 1.5 Mbps (pruebas ADSL).
- Conexión entre el equipo de prueba y un equipo asociado al GGSN 3G (pruebas UTRAN).

Todas estas pruebas se realizan tanto en el enlace de bajada como en el de subida, permitiendo utilizar tráfico TCP o UDP y capturando los paquetes en ambos extremos de la conexión a los efectos de determinar las tasas de arribo, pérdidas, retardos, etc.

En la figura 7.16 se brinda además un detalle de una red de acceso celular típica, tanto 3G como 2G, detallando la ubicación de los equipos de interés para las pruebas.

### Pruebas ADSL.

Estas pruebas se realizaron por dos motivos. Por un lado a los efectos de tener un punto de referencia de funcionamiento y por otro para verificar que las zonas de congestión no estaban fuera de la red celular.

Se realizaron pruebas de descarga FTP desde el servidor `arbolita` a través de un servicio ADSL de 1.5Mbps, para verificar la tasa de transmisión que es posible obtener a través de una conexión TCP y descartar posibles problemas en el funcionamiento de la red fuera del ámbito de la red celular.

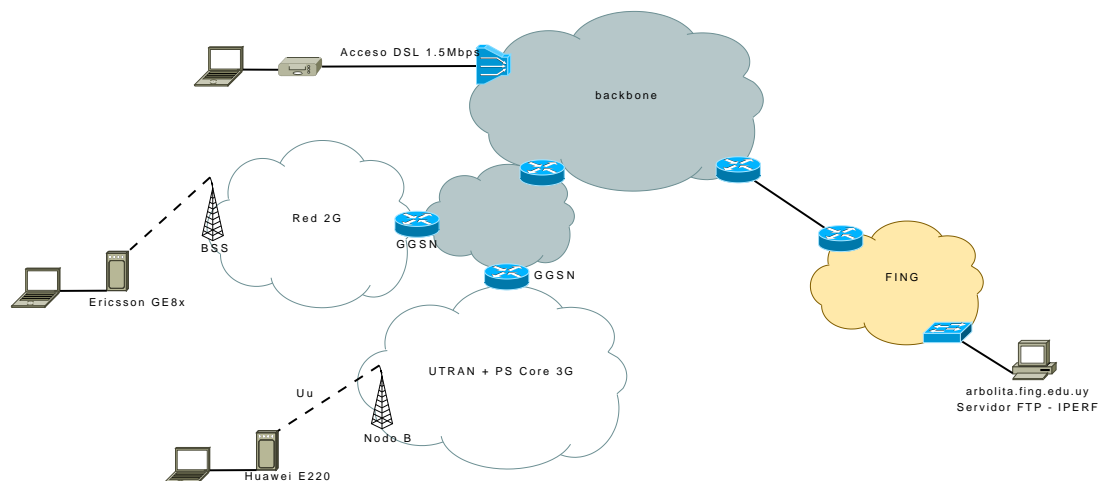


Figura 7.15: Topología de pruebas

En la figura 7.17 se grafica el número de secuencia TCP (relativo al comienzo de la conexión) en función del tiempo, para una de las tantas pruebas realizadas. La pendiente de dicha gráfica proporciona una medida de la tasa de descarga obtenida. En este caso, el comportamiento es casi ideal, obteniéndose una tasa regular de descarga de 1.29Mbps.

### Pruebas 3G.

Se describen a continuación los resultados de las diferentes pruebas realizadas sobre la red 3G. Para todas estas pruebas se utilizó un modem HSDPA Huawei E220 conectado vía USB a una PC con sistema operativo Windows. Las pruebas que se realizaron son de dos grandes tipos:

- **Pruebas TCP:** descargas FTP (sobre TCP) desde el servidor *arbolita* situado en la red de FING hacia el PC con modem. En estas pruebas se realizó captura de paquetes tanto del lado del servidor FTP como del PC cliente a los efectos de analizar la tasa y el orden de arribos, las variaciones de retardo entre paquetes, el comportamiento de la ventana de transmisión TCP, retransmisiones, timeouts, tanto en el enlace de bajada (datos) como en el de subida (reconocimientos TCP o ACKs).
- **Pruebas UDP:** Aquí se probó enviar paquetes UDP a tasa constante de diferentes tamaños de manera de determinar la capacidad del cuello de botella, así como determinar posibles problemas de congestión, y de evaluar el funcionamiento de la capa 2. Se realizaron envíos desde *arbolita* al PC de pruebas y viceversa, utilizando la herramienta iperf y el programa MetroRed que se describirá en el capítulo 10. Esto permite realizar mediciones y detectar problemas tanto en el enlace de subida (UMTS) como en el de bajada (HSDPA).

A su vez, estas pruebas se realizaron en diferentes horarios y ubicaciones, de manera de capturar diferentes situaciones de la red, como ser las condiciones de radiofrecuencia (RF) y congestión. Se describen a continuación los resultados obtenidos.

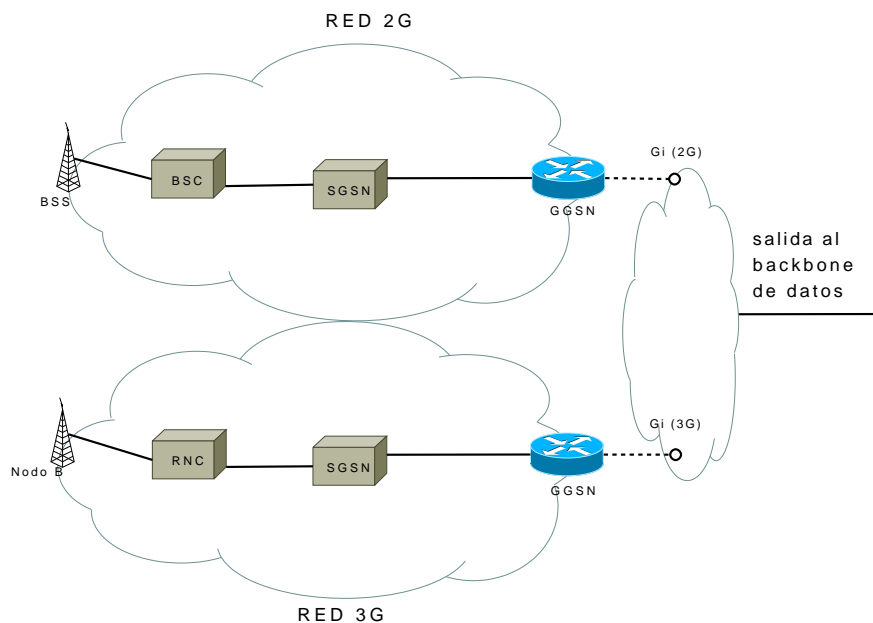


Figura 7.16: Red celular de agregación

### Pruebas TCP

Se realizaron diferentes pruebas de descarga TCP vía el modem Huawei E220. A continuación se documentan dos pruebas, con resultados representativos del comportamiento de la red HSDPA. Los detalles de la primera prueba se muestran en el cuadro 7.9. A modo de comparación con la figura

Tecnología	HSDPA
Fecha	3/3/2008
Hora	8:00
Lugar	IIE-FING
Equipos	Servidor arbolita → Huawei E220
Protocolo	TCP

Cuadro 7.9: Prueba 1.

7.17, se grafica el no. de secuencia recibido en función del tiempo. El resultado puede verse en la figura 7.18.

En este caso, el throughput medio obtenido en la conexión, sobre un total de  $\approx 9Mbytes$  transferidos es de  $600kbps$ . Sin embargo, estudiando la figura 7.19, donde se grafica el throughput instantáneo de la conexión, se observa que el mismo llega a picos de  $1,5Mbps$ . El valor de  $1,5Mbps$  era un valor razonable en esta etapa de instalación por limitaciones de los enlaces entre el RNC y la GGSN que se estaban utilizando en ese momento que limitaban a menos de 2 Mbps la capacidad

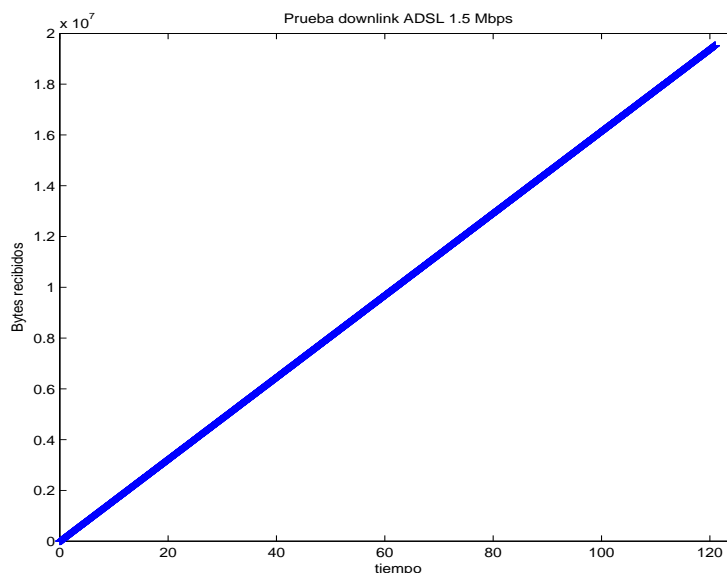


Figura 7.17: Conexión FTP a través de un servicio ADSL (bajada)

Tecnología	HSDPA
Fecha	3/3/2008
Hora	15:17
Lugar	IIE-FING
Equipos	Servidor <i>arbolita</i> → Huawei E220
Protocolo	TCP

Cuadro 7.10: Prueba 2.

del camino. Es decir, el sistema estaba proveyendo por momentos el ancho de banda deseado, aún con una única conexión TCP establecida. Este throughput a diferencia del caso GPRS no presenta fuertes variaciones si se agregan más conexiones simultáneas. Sin embargo el throughput medio es sensiblemente bajo. Sin embargo, a lo largo de una conexión el sistema frecuentemente queda bloqueado. De las capturas surge que esto sucede cuando el servidor (*arbolita*) ha agotado su ventana de transmisión y por lo tanto espera la llegada de ACKs para continuar transmitiendo. Se constató que estos bloqueos se deben a que los ACKs provenientes de la PC de pruebas sufren retardos bastante mayores a los  $100ms$  promedio para la configuración de prueba. Esto bloquea la conexión, pero además altera la estimación de retardos en el servidor y comienza a provocar timeouts que obligan a éste a disminuir la ventana, como se observa en la figura 7.20.

Se realizó una segunda prueba en una hora diferente con los detalles que se muestran en el cuadro 7.10.



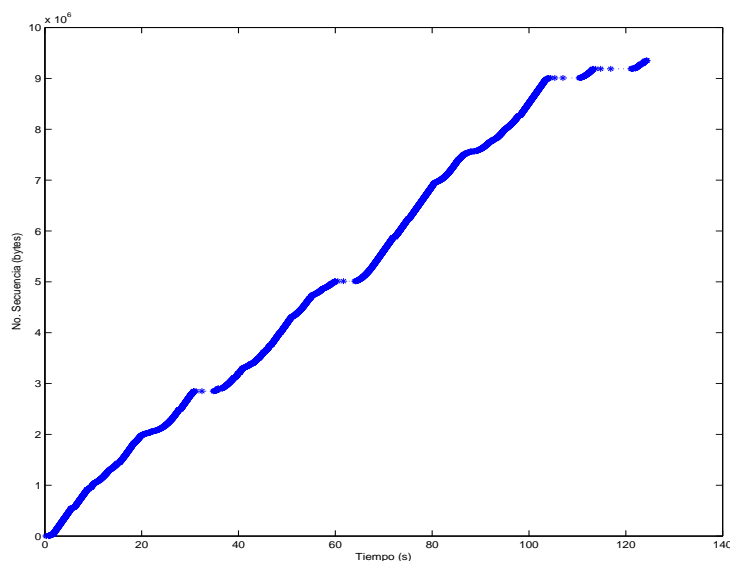


Figura 7.18: Bajada TCP en HSDPA

Como se observa en las figuras 7.21, 7.22 y 7.23, en este caso el throughput obtenido es bastante menor, con un promedio a lo largo de la conexión de  $\approx 50\text{kbps}$ , y picos de  $400\text{kbps}$ . Varias pruebas realizadas en este horario arrojaban valores medios throughput entre menos  $50\text{kbps}$  y  $200\text{kbps}$  pero raramente valores superiores. Se debe tener en cuenta que en esta etapa la red tenía aún pocos usuarios debido a que recién se estaba implantando. En este caso, el fenómeno de bloqueo mencionado anteriormente se vuelve mucho más pronunciado, con bloqueos de hasta 10 segundos y más frecuentes que en el caso anterior. A su vez, la congestión y/o las condiciones de RF en la celda parecen no permitir alcanzar el pico esperado para HSDPA.

A esta altura del análisis se puede ver que de las pruebas realizadas surge que la combinación del protocolo TCP con la interfaz de aire HSDPA presenta problemas. En primer lugar, al buscarse un throughput mucho mayor que el deseado para tecnologías anteriores (GPRS, EDGE o UMTS), es necesario que tanto el enlace de subida como el enlace de bajada funcionen correctamente. Parecen existir dos causas diferentes de problemas según lo detectado en las pruebas anteriores. Por un lado, existe una pérdida de desempeño de TCP asociada con la variación del retardo en el camino de subida UMTS por el que viajan los ACKs. Además surge que en ciertos horarios se dan condiciones de fuerte congestión en la red (ya sea por la presencia de muchos usuarios en la celda o por congestión a nivel de la red que conectaba durante esta etapa la red celular con Internet) que ocasiona fuertes problemas de desempeño. Cuánto afecta cada uno y cuál es la causa de estos problemas es el objetivo siguiente de este análisis.

Para estimar el impacto del primer problema, se realizan pruebas UDP en el enlace de subida que se documentan a continuación.

*Pruebas UDP*

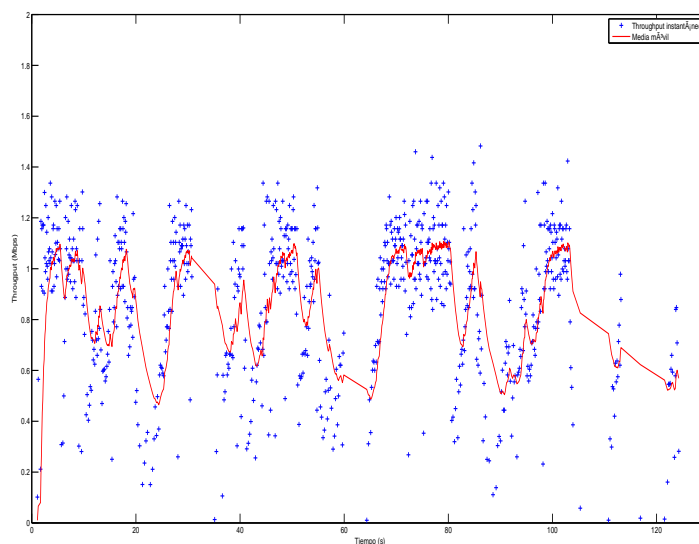


Figura 7.19: Throughput de bajada TCP en HSDPA

Tecnología	UMTS
Fecha	5/3/2008
Hora	7:30
Lugar	IIE-FING
Equipos	Huawei E220 → Servidor arbolita
Protocolo	UDP

Cuadro 7.11: Prueba 3.

Para estimar la variabilidad de retardos en los ACKS se realizaron pruebas en el enlace de subida (UMTS) desde el Huawei E220 hacia **arbolita**. Estas pruebas consisten en trenes de paquetes UDP de igual tamaño que se envían de manera de que temporalmente el enlace de subida se congestione.

Al llegar a destino, la diferencia de tiempos entre paquetes permite estimar el ancho de banda del cuello de botella y la variabilidad de retardos obtenida.

La prueba fue realizada en las mismas condiciones que la Prueba 1 de TCP como se puede observar en el cuadro 7.11.

En la figura 7.24 se observa el resultado obtenido.

Se observa que en el enlace de subida UMTS progresa en velocidad de  $64kbps$  a  $128kbps$  y por último a  $384kbps$ . Esta situación era de esperar por la configuración de la red. Esto se evidencia porque la mayoría de los paquetes presentan tiempos entre arribos correspondientes a dichas capacidades y se pueden ver los tres escalones correspondientes a lo largo de la prueba.

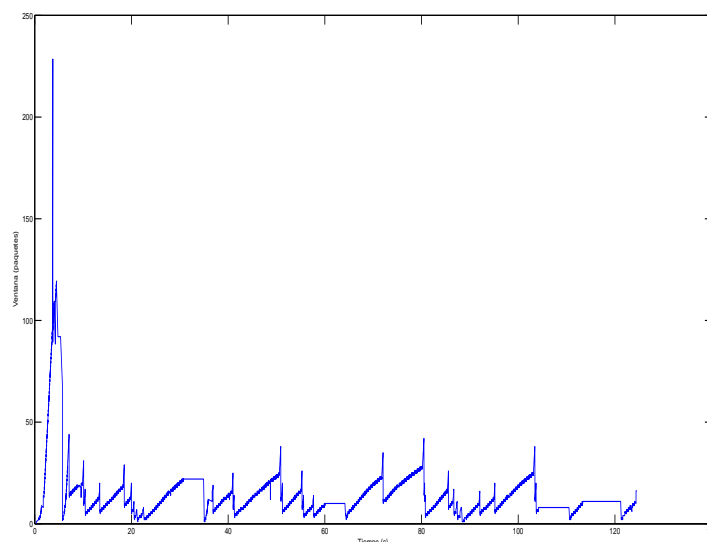


Figura 7.20: Ventana TCP en el servidor

Sin embargo, al llegar a  $384\text{kbps}$  se observa también que hay variaciones de retardo ocasionales del orden de 3 veces el retardo promedio, con picos de variación de  $400\text{ms}$  (con respecto a los  $100\text{ms}$  de media). Esto evidencia que el enlace de subida UMTS retarda algunos paquetes un tiempo bastante mayor que el habitual. Si esto le ocurre a los ACKs de una conexión TCP se producirán los bloqueos antes descritos.

En la figura 7.24 se observa además una pista de por qué esto ocurre. Asociado a cada pico de retardo hay una ráfaga de paquetes que llegan casi al unísono (con tiempos entre paquetes casi nulos). Este comportamiento no parece ser debido a problemas de congestión. Si un enlace se congestiona el tiempo entre paquetes presenta variaciones aleatorias dependiendo de cuánto tráfico cruzado se intercala entre dos paquetes de prueba sucesivos. Un ejemplo de congestión temporal se puede observar en la figura 7.25, donde se observa primero que el tiempo entre paquetes es más o menos constante y luego durante un período se intercala tráfico cruzado y se observan variaciones aleatorias del tiempo entre paquetes. Sin embargo, el fenómeno observado en el enlace de subida UMTS a  $384\text{kbps}$  es diferente. El tiempo entre paquetes no presenta variaciones importantes hasta que se da un tiempo entre paquetes muy grande y a continuación una ráfaga de varios paquetes con tiempo entre ellos casi nulo. Si bien para tener una confirmación de la causa de este fenómeno habría que tener un análisis de los paquetes en el aire, la hipótesis que surge como más razonable parece ser la siguiente. El problema se encuentra en la capa 2 de UMTS, donde fueron necesarias varias retransmisiones (entre el Modem y el RNC) para recibir apropiadamente el primer paquete de esta ráfaga (y por eso este se retarda), pero luego los paquetes son enviados en orden hacia la capa de

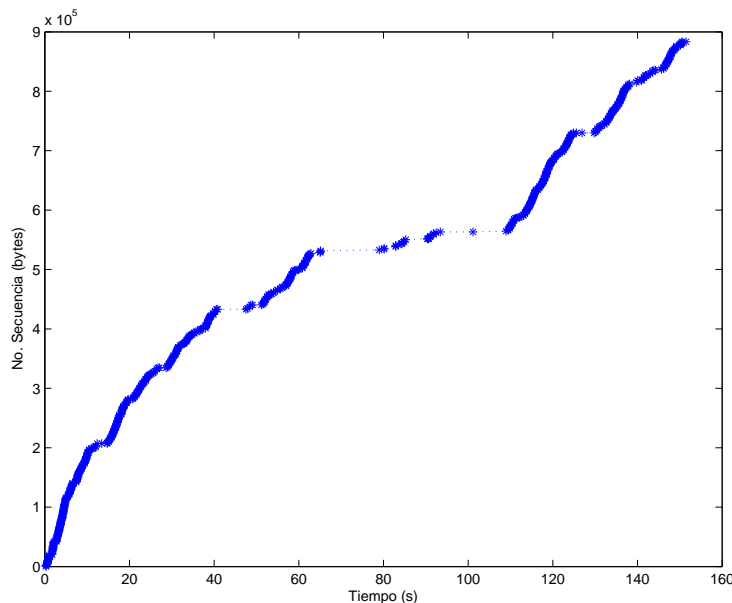


Figura 7.21: Bajada TCP en HSDPA

red, y en bloque, lo que produce este efecto. Si esto ocurre con los ACKs de una conexión TCP de bajada, el servidor recibirá conjuntos de ACKs en bloque, lo que tendrá dos efectos: en primer lugar, el retardo del primer ACK producirá un bloqueo de la conexión TCP, ya que no podrá correr su ventana de transmisión hasta recibir el ACK que se encuentra esperando. En segundo lugar, la llegada de ACKs en bloque producirá una ráfaga de paquetes en el sentido descendente de la conexión (básicamente 2 por cada ACK, debido a que normalmente se envía un ACK cada dos paquetes) que aumentará las pérdidas y congestión de la conexión en sentido descendente, con la consiguiente alta probabilidad de tener que retransmitir varios de esos paquetes y reducir la ventana.

El efecto neto entonces de los retardos variables observados en el sentido ascendente (UMTS a 384kbps) será nocivo para las condiciones TCP descendentes. Obsérvese que esto es inherente a la tecnología usada (UMTS) y no a una congestión a nivel del enlace de subida, ya que tanto 128kbps como 384kbps son suficientes en términos de ancho de banda para enviar los ACKs necesarios.

En la figura 7.26 se presenta una prueba similar pero en condiciones de una hora de mayor tráfico (15:00hs). Aquí los picos de variación de retardo llegan al orden de 1s, es decir 10 veces el valor de media.

### 7.3.3. Pruebas UTRAN.

Para identificar exactamente los problemas en el desempeño de TCP, se realizaron pruebas similares a las anteriores contra un equipo directamente conectado al GGSN, a nivel de la interfaz Gi. Esto permite separar los problemas, e identificar en cuánto afecta el desempeño el rendimiento

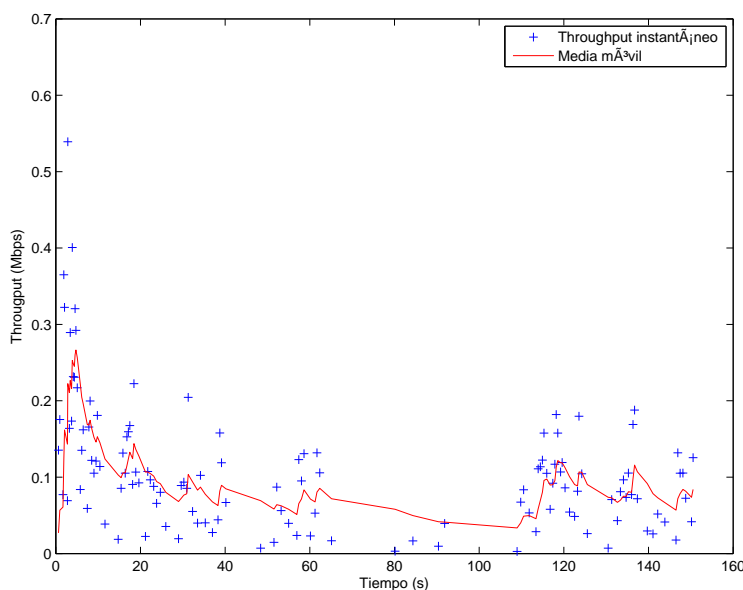


Figura 7.22: Throughput del Bajada TCP en HSDPA

de la UTRAN y los equipos de salida a internet en el Packet Switched Domain. Permite por lo tanto diferenciar qué factores afectan la baja de desempeño a nivel de Interfaz de Aire, UTRAN y SGSN/GGSN, y cuánto afectan los posibles problemas de congestión en la red que conecta la interfaz Gi con la salida a Internet.

Se realizó entonces la prueba que se muestra en el cuadro 7.12, similar a la prueba 2.

Los resultados de número de secuencia, evolución de la ventana de transmisión y throughput pueden verse en las figuras 7.27, 7.28 y 7.29. En este caso la latencia promedio de la conexión fue de  $200ms$ , y como se observa en 7.29 la evolución de la ventana TCP en este caso permite alcanzar la ventana máxima de transmisión ( $\approx 40$  paquetes).

El hecho de que esta prueba se haya realizado en condiciones similares a la prueba 2, con

Tecnología	HSDPA
Fecha	15/4/2008
Hora	15:15
Lugar	IIE-FING
Equipos	Huawei E220 → Servidor en interfaz Gi
Protocolo	TCP

Cuadro 7.12: Prueba 4.

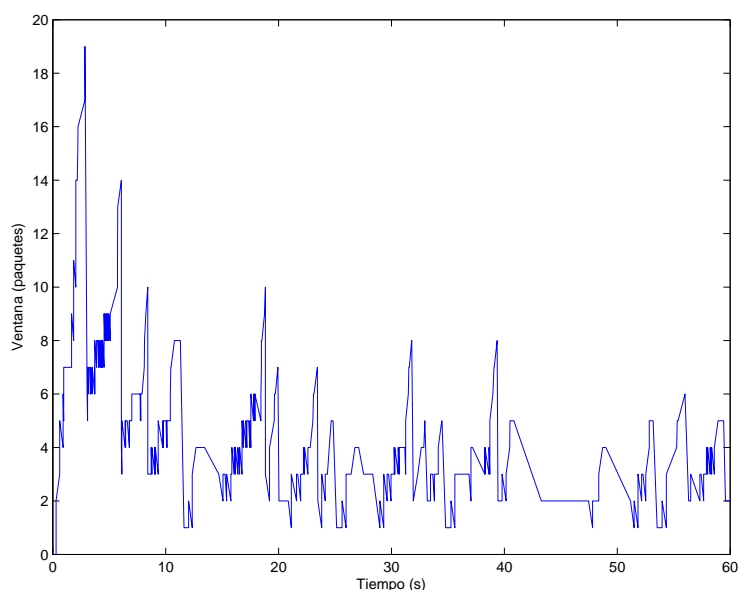


Figura 7.23: Ventana TCP en el servidor

resultados mucho mejores ( $50kbps$  promedio en la prueba 2 contra  $1,2Mbps$  promedio en esta prueba), indica que es altamente probable que la congestión en la red que conecta la interfaz Gi con la salida a Internet esté afectando fuertemente el desempeño de las conexiones TCP en las horas de fuerte demanda. Se realizaron dos pruebas similares a la prueba 4, pero contra el servidor *arbolita* y el servidor *inno.com.uy*, que está situado en un Collocation-Site de alta capacidad ambos fuera de la red celular. En ambos casos el desempeño fue similar a la prueba 2, es decir, un throughput muy por debajo de la capacidad de HSDPA. Esto indica que el problema es altamente probable que se encuentre a nivel de la red que conecta la interfaz Gi de la red celular con la salida a Internet.

Por último, se realizó una prueba UDP para nuevamente evaluar las variaciones de retardo en el enlace de subida. Esta prueba es similar a la prueba 3 y sus características se presentan en el cuadro 7.13.

En la figura 7.30 puede verse el tiempo entre arribos de paquetes, donde se observa un comportamiento similar al de la prueba 3. Se observan tiempos de arribo correspondientes a  $64kbps$ , luego  $128kbps$  y por último  $384kbps$ . Al llegar a esta tasa de transmisión, aparece nuevamente el fenómeno de variabilidad de retardos ya mencionado, con picos de  $500ms$ , es decir 5 veces el retardo promedio de la conexión. Nuevamente se observa que luego de cada paquete retrasado, llega una ráfaga de paquetes.

Esta prueba parece confirmar que el fenómeno de variabilidad de retardos en el enlace de subida

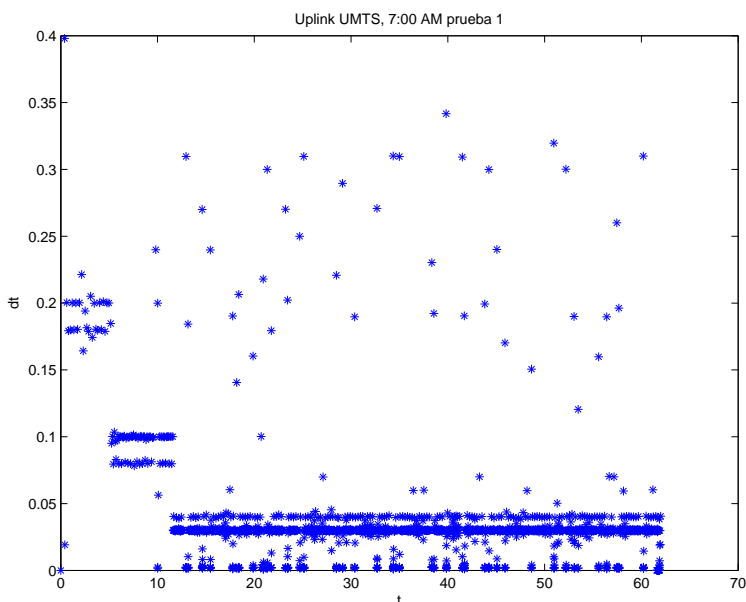


Figura 7.24: Tiempo entre arribos en UDP en el enlace de subida, prueba 1

es inherente a la red 3G y no es producto de congestión a nivel de la UTRAN o de la red de conexión a Internet, sino del mecanismo de retransmisión UMTS. Es de esperar que este fenómeno mejore con la introducción de HSUPA en el enlace de subida. Sin embargo, a modo de conclusión preliminar, puede decirse que este fenómeno resulta más significativo cuando el enlace de subida funciona a  $384\text{kbps}$ .

En base a estas pruebas se recomendó analizar con detalle la situación de congestión existente en la red que conecta la interfaz Gi con la salida a Internet que es quien afecta principalmente en esta etapa de implantación de la red el desempeño de las conexiones TCP. En segundo lugar se recomendó que para descargas vía HSDPA, el desempeño de la red es probable que mejore si se limita su conexión de subida a  $128\text{kbps}$ . Se observa que para realizar una descarga a velocidad  $2\text{Mbps}$  se necesita recibir alrededor de 160 paquetes por segundo. Para poder hacer ACK de ellos, se necesitan 80 ACKs (1 cada dos paquetes, que es el comportamiento estándar de TCP) de tamaño 60 bytes por segundo, lo que da una tasa de  $38,4\text{kbps}$  en el enlace de subida. Por lo tanto,  $128\text{kbps}$  es más que suficiente y a esa velocidad el impacto de las ráfagas de ACKs es menos notorio. Ambas sugerencias se implementaron y se logró solucionar los problemas detectados.

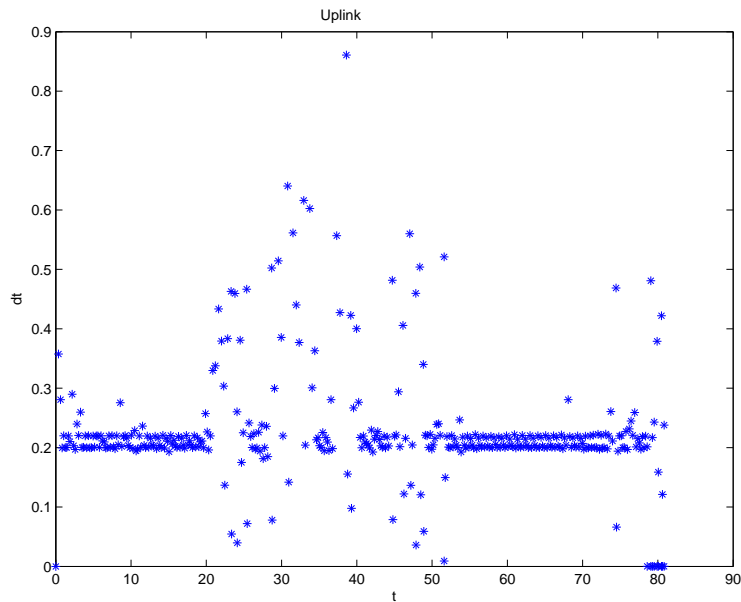


Figura 7.25: Casos de congestión temporal en el enlace de subida

## 7.4. Conclusiones

En este capítulo se realizó en primer lugar una breve reseña de trabajos en el área de mediciones activas. Posteriormente, se analizaron las dificultades que presenta estimar la capacidad de un enlace celular y se propuso un método para hacerlo. Mediante mediciones realizadas en una red operativa se observó que los resultados obtenidos son consistentes. Luego, se analizó el desempeño de TCP sobre enlaces celulares. Por un lado, en el caso de redes *GPRS/EDGE* se analizó el alto impacto que tienen los algoritmos de corrección de errores de capa 2 sobre el desempeño de TCP. Estos algoritmos corrigen errores pero agregan retardo y eso tiene un impacto muy fuerte sobre TCP. Por último, se propuso una arquitectura y una metodología de pruebas que permite detectar problemas de desempeño en redes 3G/HSDPA y evaluar si las causas de estos problemas se encuentran en la red de acceso de radio frecuencia o si se deben a fenómenos de congestión en la red de datos de salida a Internet. En el caso de estudio en que se probó este procedimiento, se detectaron los problemas existentes en una red HSDPA, en base a los resultados de las mediciones se propusieron soluciones y estas al implementarse, validaron el análisis realizado con la metodología propuesta.



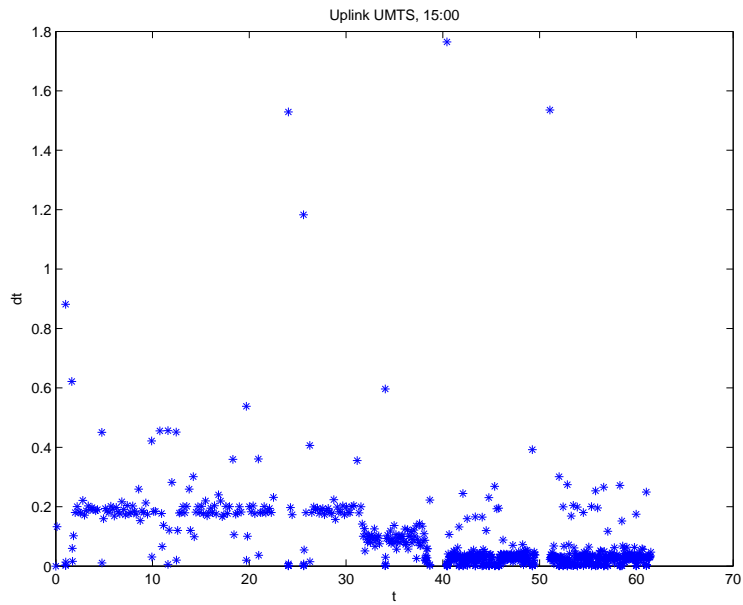


Figura 7.26: Tiempo entre arribos en UDP en el enlace de subida, prueba 2

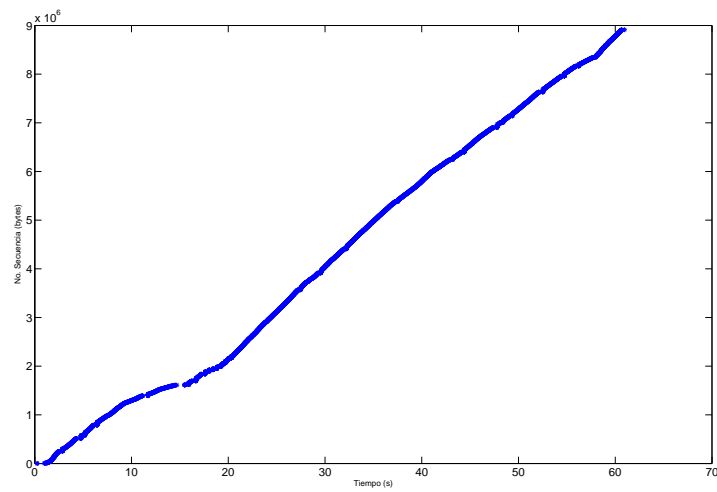


Figura 7.27: Bajada TCP en HSDPA, interfaz Gi

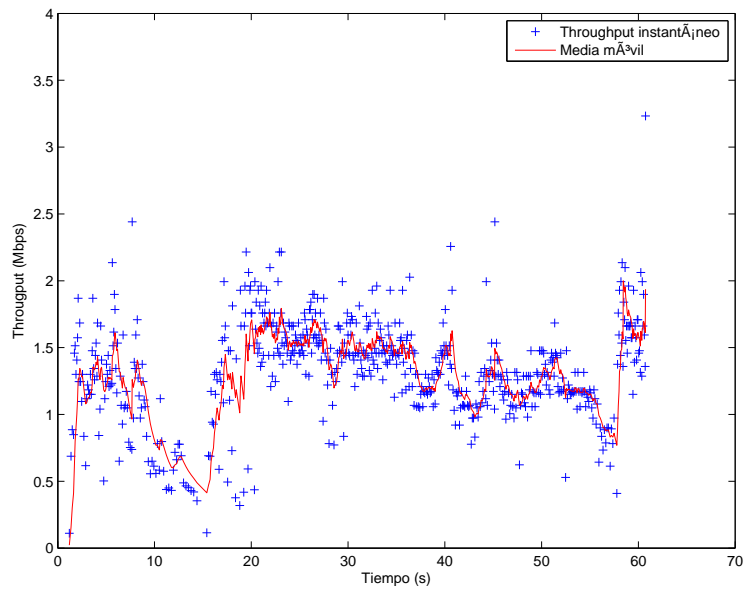


Figura 7.28: Throughput de bajada TCP en HSDPA contra interfaz Gi

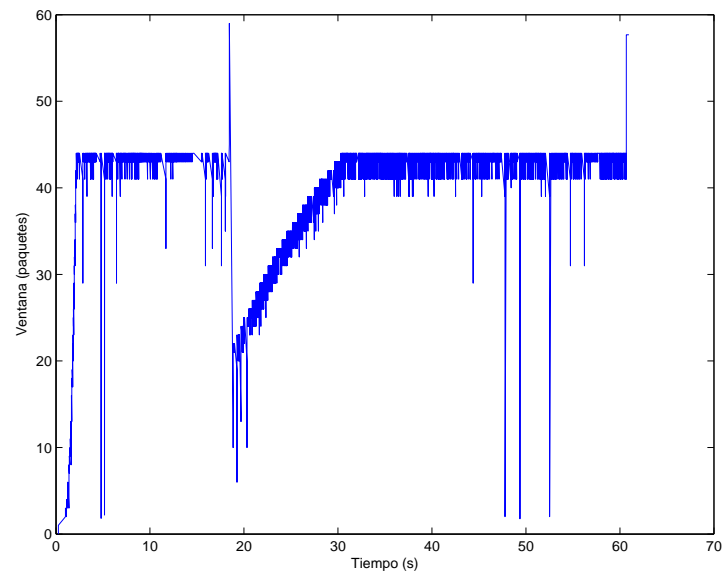


Figura 7.29: Ventana TCP en el servidor de la interfaz Gi

Tecnología	UMTS
Fecha	14/4/2008
Hora	14:45
Lugar	IIE-FING
Equipos	Huawei E220 → Servidor en interfaz (Gi)
Protocolo	UDP

Cuadro 7.13: Prueba 5.

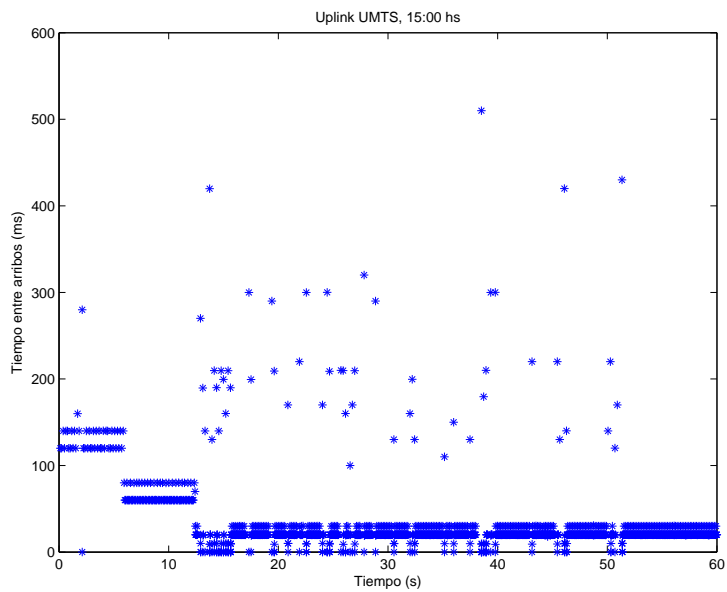


Figura 7.30: Tiempo entre arribos en UDP en la subida UMTS contra Interfaz Gi

## Capítulo 8

# Aprendizaje estadístico y su aplicación al monitoreo y control de admisión en redes

### 8.1. Introducción

El objetivo de este capítulo y el siguiente es diseñar un sistema de monitoreo de la calidad de servicio de extremo a extremo del tráfico que genera una aplicación en una red. Este sistema de monitoreo puede contribuir a la implementación de diferentes herramientas tanto para un operador como para los usuarios. Para un operador puede permitirle realizar control de admisión en una red de servicios premium, analizar cómo evoluciona a lo largo del día la QoS de ciertas aplicaciones de su red, analizar tendencias en la QoS a lo largo del tiempo con fines de planificación, etc.. Una posible aplicación de interés, como se mencionó antes, es controlar la admisión de nuevas conexiones de ciertos servicios o aplicaciones. Con un sistema de este tipo, una aplicación puede decidir si es posible o no admitir una nueva conexión de la aplicación de interés teniendo en cuenta la predicción de la QoS de extremo a extremo que obtendría esa nueva conexión si fuera admitida.

Para una aplicación del usuario, este sistema de monitoreo podría permitirle por ejemplo hacer un sistema de control para adaptar las aplicaciones al estado de la red, o en el caso de multihoming decidir por donde enviar ciertas aplicaciones o monitorear un acuerdo de servicio (SLA) con un operador.

Un aspecto en común de las diferentes aplicaciones reseñadas, es que en todos los casos interesa monitorear la calidad de servicio durante largos períodos de tiempo.

La forma más simple de monitorear la calidad de servicio de extremo a extremo de una aplicación es enviar el tráfico de dicha aplicación (un video por ejemplo) y sobre él medir los parámetros de calidad de servicio. Esto en muchas aplicaciones no es posible o razonable. Por ejemplo, en servicios como el video de buena definición, el ancho de banda que consume la aplicación es de una magnitud tal que puede representar un porcentaje no despreciable del ancho de banda de uno más de los enlaces del acceso a la red (en uno o ambos extremos del camino a monitorear). Por lo tanto, enviar un video para evaluar la QoS, si la red se encuentra algo cargada puede implicar deteriorar

la calidad de servicio de los usuarios que ya están usando la red. Este mecanismo lo usan diversos equipos de monitoreo como por ejemplo aplicaciones comerciales de CISCO o Telgat, este último en el caso de redes celulares.

Una alternativa para evitar cargar la red con el sistema de medidas (y que también se utiliza actualmente en algunas aplicaciones de monitoreo), es enviar algunos paquetes de tamaño pequeño y que no carguen la red. Sobre estos paquetes de prueba, se mide por ejemplo el retardo que reciben y con esa medida se extrapola al desempeño de la aplicación de interés. El problema de esta técnica es que esta última extrapolación en general no brinda información adecuada. Los paquetes de prueba pueden sufrir muy poco retardo y la aplicación (un video por ejemplo) sufrir un retardo importante dependiendo de la estadística del video y del estado de la red.

En este trabajo se busca un punto intermedio entre las dos aproximaciones anteriores. La idea será enviar paquetes de prueba livianos, con ellos estimar el estado de la red en ese momento y luego estimar mediante técnicas de aprendizaje estadístico cuál será la QoS que recibiría el tráfico de la aplicación si hubiera sido enviado en ese momento. Esta técnica de monitoreo, luego de una fase inicial de aprendizaje, permite cargando poco la red obtener una buena aproximación de la calidad de servicio que recibiría una aplicación si fuera enviado por ese camino.

Otra forma de analizar el problema de estimar la QoS de una aplicación, se puede ver en el trabajo de Tao y Guérin [140]. En este trabajo los autores analizan la estimación de las pérdidas de extremo a extremo que ve una aplicación. En dicho trabajo se busca caracterizar las pérdidas que ve la aplicación a partir de las pérdidas que sufre el tráfico de prueba. Se plantea que el proceso de pérdidas que sufre la aplicación corresponde a un “hidden markov model” (HMM) con dos estados. Por lo tanto, lo que se hace es ajustar los parámetros del modelo HMM a partir de la información de pérdidas que se observa en el tráfico de pruebas. En el citado trabajo se asume además que el tráfico de prueba y la aplicación ven la misma probabilidad de pérdidas (esto significa que ni el tráfico de prueba ni la aplicación afectan la probabilidad de pérdida del camino). Además, el análisis realizado es válido bajo la hipótesis de que la aplicación pueda ser modelada como un tráfico tipo ON-OFF. Las diferencias principales de dicho trabajo con la propuesta de esta tesis son: el modelo de esta tesis es más general en varios aspectos: no vale sólo para pérdidas sino también para otros parámetros de desempeño del camino, utiliza más información de los paquetes de pruebas que las pérdidas, no asume un modelo tan específico para el proceso de pérdidas sino que se basa en aprender la función que relaciona el parámetro de desempeño de interés con las modificaciones que experimenta el tráfico de prueba al pasar por el camino y por último, no asume un modelo específico para el tráfico de la aplicación ni que este tráfico y el de prueba deban experimentar la misma probabilidad de pérdida de paquetes ni de ningún otro parámetro de QoS.

El análisis de este problema y de las soluciones propuestas se realizará en este capítulo y en el siguiente. En este capítulo se analizará en la sección 8.2 una técnica de aprendizaje estadístico utilizando el estimador de Nadaraya-Watson. Posteriormente en la sección 8.3, se propondrá la extensión para el caso no estacionario de los resultados existentes en la literatura para el estimador de Nadaraya-Watson funcional. Por último en la sección 8.4, se propondrá una metodología que aplicando el resultado anterior para el estimador de Nadaraya-Watson permite estimar la QoS de extremo a extremo de una aplicación, utilizando como estimador del estado de la red la función distribución empírica del tiempo entre arribos de paquetes de prueba consecutivos.

## 8.2. Aprendizaje estadístico

### 8.2.1. Introducción a los modelos de regresión

En el modelo de regresión se representa la relación entre dos variables aleatorias  $X$  e  $Y$  mediante la ecuación

$$Y = \phi(X) + \varepsilon,$$

donde  $\varepsilon$  es una variable aleatoria, centrada e independiente de  $X$  que representa los errores y  $\phi$  es una función no aleatoria. Este modelo incluye varios problemas diferentes. Uno de los problemas consiste en conocer la función  $\phi$ . Esta función se puede estimar a partir de observaciones de las variables  $(X, Y)$ . Luego mediante nuevas observaciones se puede validar el modelo o también se pueden hacer predicciones sobre la variable  $Y$ , es decir que conociendo  $X$  y la estimación de  $\phi$  se estiman futuros valores de  $Y$ .

Existen diferentes técnicas de regresión. Un posible criterio de clasificación es entre técnicas de regresión paramétrica y técnicas no paramétricas. En las técnicas paramétricas se asumen modelos para la función a estimar. La técnica más conocida es la regresión lineal donde se asume un modelo lineal para la función y se estiman los parámetros de esta función. En el caso no paramétrico se hacen hipótesis generales sobre la función a estimar (continuidad, derivabilidad, etc.) pero la función no se determina por un conjunto finito de parámetros.

En la tesis se analizarán dos técnicas de regresión: Nadaraya-Watson (NW) y Support Vector Machines (SVM). La primera cae dentro de la clasificación de regresión no paramétrica. La segunda es una técnica en cierto sentido mixta, ya que si bien asume un modelo lineal, este modelo lineal se realiza en un espacio de alta dimensión por lo que se adapta muy bien para estimar funciones bajo hipótesis muy generales. Además, no requiere almacenar una cantidad de parámetros del orden de la dimensión del espacio como los modelos paramétricos tradicionales.

Otro aspecto a considerar al estudiar regresión es el espacio al que pertenecen las variables  $X$ . En este trabajo se analizará el caso en que las variables  $X$  pertenecen a  $\mathbb{R}^d$  y el caso en que pertenecen a un espacio de funciones.

Una referencia para una introducción general al problema de regresión no paramétrica en el caso de variables en  $\mathbb{R}^d$  es el libro de Nadaraya [108]. Las referencias para el caso funcional se encuentran en diversos artículos [54, 55, 56, 99, 112].

En el caso de Support Vector Machines una buena referencia es el libro de Vapnik [143].

### 8.2.2. El estimador de Nadaraya-Watson para variables en $\mathbb{R}^d$

Se considera una variable aleatoria  $X$  en  $\mathbb{R}^d$ , una variable aleatoria  $Y$  en  $\mathbb{R}$ , una función  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$  y la regresión  $Y = \phi(X) + \varepsilon$ , con  $\varepsilon$  otra variable aleatoria en  $\mathbb{R}$ , centrada e independiente de  $X$ .

El estimador de Nadaraya-Watson fue introducido por ambos autores [107, 155]:

$$\phi_n(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{x-X_i}{h_n}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right)}$$

si  $\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right) \neq 0$  y si  $\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right) = 0$  se define  $\phi_n(x) = 0$ .  $K$  es un núcleo, es decir una función  $K: \mathbb{R}^d \rightarrow \mathbb{R}$  que cumple que  $K(x) \geq 0 \forall x \in \mathbb{R}^d$  y  $\int_{\mathbb{R}^d} K(x)dx = 1$ , y  $h_n$  es una sucesión de reales positivos, que tiende a 0 con  $n$ , que se denomina ventana.

Una interpretación de la fórmula del estimador se puede obtener si se consideran variables discretas. Para estimar  $E(Y|X)$  basta estimar  $\phi(x) = E(Y|X = x)$  para  $x$  en el recorrido de  $X$ . En este caso  $E(Y|X = x)$  se puede estimar con el promedio de los  $Y_i$  tales que  $X_i = x$ , es decir,

$$\phi_n(x) = \sum_{i=1}^n Y_i 1_{\{X_i=x\}}$$

Quando la variable  $X$  no es discreta si se estima  $\phi(x) = E(Y|X = x)$  promediando los  $Y_i$  tales que  $X_i$  está “cerca” de  $x$  y se obtiene un estimador de la forma

$$\phi_n(x) = \sum_{i=1}^n Y_i w_n(X_i, x)$$

donde  $w_n(X_i, x)$  son “pesos” que indican si los datos están “cerca” o “lejos” de  $x$  para alguna medida de distancia definida en el espacio de las variables  $X$ . En este caso los pesos están determinados por

$$w_n(X_i, x) = \frac{K\left(\frac{x-X_i}{h_n}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right)},$$

donde la ventana regula el peso que tendrá cada dato en la estimación.

Se denotan las siguientes expresiones como  $g_n(x)$  y  $f_n(x)$

$$g_n(x) = \frac{1}{nh_n^d} \sum_{i=1}^n Y_i K\left(\frac{x-X_i}{h_n}\right)$$

$$f_n(x) = \frac{1}{nh_n^d} \sum_{i=1}^n K\left(\frac{x-X_i}{h_n}\right)$$

de modo que el estimador se escribe como  $\phi_n(x) = \frac{g_n(x)}{f_n(x)}$ .

La normalización de las variables  $f_n(x)$  y  $g_n(x)$  se realiza mediante  $nh_n^d$ , es decir que depende de la dimensión del espacio, más concretamente del volumen de la bola de centro 0 y radio  $h_n$ . Esta normalización es necesaria para probar la convergencia de las funciones anteriores.

El problema de la dimensionalidad del espacio sobre el que se está estimando, es decir que para estimar en dimensiones mayores son necesarios más datos porque la concentración de una cantidad fija de datos disminuye al aumentar la dimensión, queda reflejado en esta normalización. Este punto deberá ser tenido en cuenta cuando se analice el caso de regresión en espacios de dimensión infinita (cuando se quiera estimar en espacios funcionales) porque será necesario introducir alguna normalización que permita la convergencia.

Para este estimador, si se asume que existen densidades se demuestra la convergencia en probabilidad, es decir que

$$\phi_n(x) \xrightarrow{p} \phi(x),$$

cuando  $n \rightarrow \infty$ , donde  $\xrightarrow{p}$  indica convergencia en probabilidad.

También se prueba convergencia y normalidad asintótica, es decir que

$$\sqrt{nh_n}(\phi_n(x) - \phi(x)) \xrightarrow{w} N(0, \sigma^2(x)),$$

donde  $\xrightarrow{w}$  indica convergencia en distribución y  $N(0, \sigma^2(x))$  representa una variable con distribución gaussiana con media 0 y varianza  $\sigma^2(x)$ . Bajo diferentes hipótesis de momentos para las variables  $(X, Y)$  y diferentes velocidades de convergencia a 0 de la ventana se demuestran estos resultados. En todos los casos la velocidad de convergencia a 0 de la ventana debe ser pequeña comparada con la velocidad con la que  $n$  tiende a  $\infty$  y además depende de la dimensión del espacio (por ejemplo en  $\mathbb{R}^d$  es necesario que  $\lim_{n \rightarrow \infty} nh_n^d = \infty$ ).

### 8.2.3. EL estimador de Nadaraya-Watson para variables funcionales

Para el caso en que, para cada  $n \geq 1$ ,  $X_n$  es una función aleatoria en un espacio de funciones  $\mathcal{D}$  con una seminorma  $\|\cdot\|$ , Ferraty, Goia y Vieu [56] introducen el estimador  $\phi_n$  para el modelo de regresión, definido por

$$\phi_n(x) = \frac{\sum_{i=1}^n Y_i K\left(\frac{\|x - X_i\|}{h_n}\right)}{\sum_{i=1}^n K\left(\frac{\|x - X_i\|}{h_n}\right)},$$

donde se considera  $0/0 = 0$ .

Ferraty et al. [56] prueban que si  $\mathcal{D}$  es un espacio vectorial de funciones con una seminorma  $\|\cdot\|$  y  $(X_n, Y_n)_{n \geq 1}$  una sucesión de variables estacionarias donde  $X_n \in \mathcal{D}$ ,  $Y_n \in \mathbb{R}$ , bajo hipótesis habituales sobre el núcleo, la dependencia débil de las variables, y la siguiente hipótesis:

**Hipótesis 8.1.** *Para cada  $x \in \mathcal{D}$  existen  $\delta(x) > 0$  y  $c(x) > 0$  tales que*

$$\lim_{h \rightarrow 0^+} \frac{P(\|X - x\| \leq h)}{h^{\delta(x)}} = c(x)$$

y para todo  $i \neq j$ , existe  $\delta_0(x) > 0$  tal que

$$\lim_{h \rightarrow 0^+} \frac{P(\|X_i - x\| \leq h, \|X_j - x\| \leq h)}{h^{\delta(x) + \delta_0(x)}} = c_{i,j}(x),$$

con  $0 < c_1(x) \leq c_{i,j}(x) \leq c_2(x) < \infty$ .

Entonces  $\phi_n(x)$  converge completamente a  $\phi(x)$  cuando  $n \rightarrow \infty$ , es decir que para todo  $\varepsilon > 0$ , se verifica que

$$\sum_{n=1}^{\infty} P(|\phi_n(x) - \phi(x)| > \varepsilon) < \infty.$$



De las varias hipótesis del teorema anterior sólo se refirió la hipótesis 8.1 que tiene especial importancia e interés para el caso funcional.

En este artículo los autores introducen la dimensión fractal ( $\delta(x)$  en la hipótesis 8.1) del proceso  $X$  para resolver el problema de dimensionalidad. Esta hipótesis se refiere a la concentración de la variable  $X$  y a la distribución conjunta de  $(X_i, X_j)$ . La hipótesis sobre dimensión fractal del proceso  $X$  garantiza la concentración de las observaciones alrededor de  $x$  necesaria para estimar  $\phi(x)$ . La normalización en el estimador para demostrar la convergencia depende de esta dimensión fractal, es decir que la normalización que para  $X \in \mathbb{R}^d$  (asumiendo densidades) es  $nh_n^d$  en este caso es  $nh_n^{\delta(x)}$  donde  $\delta(x)$  está definida en la hipótesis 8.1. Se puede interpretar que localmente el comportamiento es como en un espacio de dimensión finita, con dimensión  $\delta(x)$ . Análogamente al caso en  $\mathbb{R}^d$ , este valor, junto con las condiciones de dependencia débil, determina la velocidad de convergencia de la ventana a 0 para obtener la convergencia y las tasas de convergencia.

Masry [99] prueba también la normalidad asintótica de  $\phi_n$  para variables débilmente dependientes asumiendo estacionariedad de segundo orden e hipótesis similares a las anteriores respecto a la concentración de variables en un entorno del punto a estimar.

### 8.3. Extensiones del estimador funcional de Nadaraya-Watson al caso no estacionario

Como se vio en la sección 8.1, la motivación de este trabajo se centra en la estimación de parámetros de QoS para servicios en redes de datos como Internet. La calidad de servicio en Internet depende de las características del tráfico que atraviesa la red. Como se verá más adelante, este tráfico es típicamente no estacionario y por lo tanto para aplicar las técnicas de regresión antes vistas se deberán extender los resultados anteriores para el caso no estacionario. Este resultado fue publicado junto con la propuesta de metodología que permite aplicarlo a la estimación de la QoS de extremo a extremo en [12, 13].

Para este análisis, en lugar de considerar variables aleatorias  $X$  igualmente distribuidas se considera un modelo basado en el introducido por Perera [122] y definido por

$$X_i = \varphi(\xi_i, Z_i) \tag{8.1}$$

donde  $\xi_i$  toma valores en un espacio vectorial seminormado y  $Z_i$  es una variable aleatoria real que toma valores en un conjunto finito  $\{z_1, z_2, \dots, z_m\}$ . La sucesión  $\varphi(\xi_i, z_k)$  es débilmente dependiente e igualmente distribuida para todo  $1 \leq k \leq m$ , pero la secuencia  $Z_i$  puede ser no estacionaria.

El modelo representa un mezcla de procesos débilmente dependientes y estacionarios, pero la mezcla es no estacionaria y dependiente. Más adelante se analizará una interpretación más precisa de este modelo en el contexto de este trabajo. A los efectos de dar una idea intuitiva de la aplicación de este modelo, se puede considerar que  $\varphi(\xi_i, z_k)$  representa un tipo de tráfico de los observados, y la variable  $Z$  selecciona para cada período de tiempo el tipo de tráfico. Hay que observar que con este modelo se pueden representar los diferentes regímenes de tráfico observados en la red en diferentes escalas temporales.

En lo que sigue de esta sección se explicará el resultado que prueba la convergencia del estimador de Nadaraya-Watson funcional en el caso no estacionario. Los detalles técnicos de la demostración

para facilitar la lectura del documento, se refieren al Apéndice 8.6 o a los artículos [12, 13].

Posteriormente Aspirot, Bertin y Perera demostraron la normalidad asintótica de este estimador en [7, 8].

La consistencia del estimador se obtiene mediante argumentos similares a los de [56], condicionando a la variable  $Z$  y luego hipótesis generales para esta variable permiten obtener el resultado sin condicionar.

Para esto se calcula primero el límite de  $E(f_n(x))$  y de  $E(g_n(x))$ . Se considera

$$\phi_n(x) = \frac{g_n(x)}{f_n(x)},$$

donde,

$$g_n(x) = \frac{1}{n\psi(h_n)} \sum_{i=1}^n Y_i K_n(X_i),$$

$$f_n(x) = \frac{1}{n\psi(h_n)} \sum_{i=1}^n K_n(X_i),$$

y  $\psi(h_n)$  es una normalización que depende de la concentración de observaciones alrededor del punto a estimar  $x$  y se define con precisión más adelante.

**Lema 8.1.** *Sea  $X = (X_n)_{n \geq 1}$ ,  $Y = \phi(X) + \varepsilon$ , con  $\varepsilon$  una variable aleatoria real e independiente de  $X$  que satisface las siguientes hipótesis:*

**Hipótesis 8.2.** *Existen dos procesos independientes que satisfacen  $\xi = (\xi_n)_{n \in \mathbb{N}}$ ,  $Z = (Z_n)_{n \in \mathbb{N}}$ , tal que  $\xi$  es estacionario, con valores en un espacio funcional,  $Z$  es real con valores en  $\{z_1, \dots, z_m\}$  y existe una función  $\varphi$  que toma valores en un espacio funcional  $\mathcal{D}$  tal que*

$$X_n = \varphi(\xi_n, Z_n).$$

**Hipótesis 8.3.** *Existen funciones positivas  $\psi, \psi_1, \dots, \psi_m$  definidas en  $\mathbb{R}^+ \times \mathcal{D}$ ,  $c_1, \dots, c_k$  definidas en  $\mathcal{D}$  y un subconjunto  $\Delta \subset \{1, \dots, m\}$  tal que para todo  $h > 0$*

$$P[\|\varphi(\xi_1, z_k) - x\| \leq h] = c_k(x)\psi_k(h, x),$$

con  $\lim_{h \rightarrow 0} \frac{\psi_k(h, x)}{\psi(h, x)} = 1$  si  $k \in \Delta$ , y  $\lim_{h \rightarrow 0} \frac{\psi_k(h, x)}{\psi(h, x)} = 0$  si  $k \in \Delta^c$ , donde  $\Delta^c$  es el complemento del conjunto  $\Delta$ .

En lo que sigue se notará  $\psi(h_n)$  en lugar de  $\psi(h_n, x)$ , para simplificar la notación.

**Hipótesis 8.4.** *Las funciones  $u \mapsto \psi_k(u, x)$  son diferenciables en  $\mathbb{R}^+$ , con derivada  $\psi'_k(u, x)$  y*

$$\lim_{h \rightarrow 0} \frac{h}{\psi_k(h, x)} \int_0^1 K(u)\psi'_k(uh, x)du = d_k(x),$$

donde  $d_k$  son funciones definidas en  $\mathcal{D}$ .

**Hipótesis 8.5.** Para todo  $k \in \{1, \dots, m\}$  el siguiente límite existe

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n P(Z_i = z_k)$$

y se notará  $p_k$ .

**Hipótesis 8.6.**  $\phi$  es una función continua.

**Hipótesis 8.7.**  $K$  es positivo con soporte en  $[0, 1]$ .

**Hipótesis 8.8.** La ventana  $h_n$  satisface que  $\lim_{n \rightarrow \infty} h_n = 0$ .

Entonces :

- $\lim_{n \rightarrow \infty} E(f_n(x)) = f(x)$ , donde  $f(x) > 0$  y  $f$  esté definida para todo  $u \in \mathcal{D}$  por

$$f(u) = \sum_{k \in \Delta} p_k d_k(u) c_k(u),$$

- $\lim_{n \rightarrow \infty} E(g_n(x)) = \phi(x) f(x)$ .

**Observación 8.1.** La hipótesis 8.3, es sobre la concentración de las variables aleatorias  $\varphi(\xi_n, z_k)$ , representada por  $\psi_k(h)$ , en una bola centrada en  $x$  con radio  $h$ . En el caso real, para variables en  $\mathbb{R}^d$  con densidad, la distribución en una bola centrada en  $0$  con radio  $h$  es proporcional a  $h^d$ . Los componentes de la mezcla que finalmente determinan la normalización  $\psi$  son aquellos con índice en  $\Delta$  que corresponden a las variables con mayor concentración alrededor de  $x$ .

La hipótesis 8.4 implica que  $\|\varphi(\xi, z_k) - x\|$  tiene densidad  $c_k(x) \psi'_k$ . Para variables en  $\mathbb{R}^d$  con función de densidad,  $d_k$  es constante para todo  $x \in \mathbb{R}^d$ .

La hipótesis 8.5 garantiza estacionariedad “en media”. Esta hipótesis es verificada por ejemplo por variables periódicas. Las hipótesis 8.6, 8.7, 8.8 son usuales para la estimación por núcleos.

Para probar la convergencia casi segura se considera el estimador condicionado a los valores de la variable  $Z$ , y se trabaja con un vector aleatorio estacionario y centrado en  $\mathbb{R}^{2m}$  como se muestra a continuación.

Se considera la variable  $\tilde{X}^n = (\tilde{X}^{1,n}, \dots, \tilde{X}^{2m,n})$  con valores en  $\mathbb{R}^{2m}$  definida para  $i \geq 1$  de la siguiente manera

- para  $l \in \{1, \dots, m\}$

$$\begin{aligned} \tilde{X}_i^{l,n} &= \frac{1}{\sqrt{\psi(h_n)}} K_n(\varphi(\xi_i, z_l)) (\phi(\varphi(\xi_i, z_l)) + \varepsilon_i) \\ &\quad - \frac{1}{\sqrt{\psi(h_n)}} E[K_n(\varphi(\xi_i, z_l)) \phi(\varphi(\xi_i, z_l))], \end{aligned}$$

- para  $l \in \{m + 1, \dots, 2m\}$

$$\tilde{X}_i^{l,n} = \frac{1}{\sqrt{\psi(h_n)}} K_n(\varphi(\xi_i, z_l)) - \frac{1}{\sqrt{\psi(h_n)}} E[K_n(\varphi(\xi_i, z_l))]$$

**Teorema 8.2.** *Bajo las hipótesis del Lema 8.1 y las siguientes hipótesis adicionales*

**Hipótesis 8.9.** *Para todo  $B \subset \mathbb{N}$ ,  $n \in \mathbb{N}$ ,  $i \in \{1, \dots, 2m\}$*

$$E \left[ S_n \left( B, \tilde{X}^{i,n} \right)^4 \right] \leq C \left( \frac{\text{card}(B_n)}{n} \right)^2$$

**Hipótesis 8.10.** *La ventana  $h_n$  satisface*

$$\lim_{n \rightarrow \infty} h_n = 0$$

y para algún  $0 < \beta < 1$

$$\lim_{n \rightarrow \infty} \psi(h_n) n^{\beta/2} = \infty.$$

Entonces,  $\phi_n(x)$  converge casi seguramente a  $\phi(x)$  con  $n \rightarrow \infty$ .

**Observación 8.2.** *La hipótesis 8.9 hace referencia a la dependencia débil del proceso, y puede ser obtenida por ejemplo con hipótesis de  $\alpha$ -mixing.*

*La hipótesis 8.10 brinda la velocidad de convergencia necesaria de la ventana para probar el teorema.*

## 8.4. Estimación de la QoS de extremo a extremo de una aplicación

### 8.4.1. Análisis del problema y metodología de estimación

En esta sección se estudiará el caso de un camino con un único enlace. Consideraciones sobre el caso de un camino con múltiples enlaces se realizan en la siguiente sección.

El objetivo es estimar los parámetro de QoS (como retardo, jitter, pérdidas, etc.) de servicios de valor agregado en Internet como el video bajo demanda.

Se denominará  $Y$  al parámetro de QoS que se desea estimar. Este parámetro es función de las características del enlace y el tráfico que lo atraviesa:

$$Y = \phi(X_t, V_t, C, B)$$

donde  $X_t$  es el tráfico de los restantes usuarios que utilizan el enlace (“tráfico cruzado”),  $V_t$  es el tráfico del video sobre el que se desea estimar el parámetro de QoS,  $C$  es la capacidad del enlace, y  $B$  es el tamaño del buffer.

La capacidad del enlace  $C$  se supone que no es conocida (si bien en algún caso podría serlo o en otros se podría estimar con técnicas conocidas) y que el tamaño del buffer  $B$  tampoco lo es. También se asume que ambos son constantes durante todo el período de monitoreo de la red. El

proceso  $V_t$  es el tráfico sobre el que se quiere estimar la QoS por lo que se considera una entrada conocida para nuestro problema.

Por lo tanto se puede escribir:

$$Y = \phi(X_t) + \varepsilon$$

En la siguiente sección se analizarán las hipótesis anteriores sobre  $C$ ,  $B$ , y  $V_t$ .

En esta formulación se encuentran dos problemas diferentes. Por un lado estimar  $\phi$  que es una función desconocida, y por otro estimar el proceso  $X_t$  que también es desconocido.

Para estimar el tráfico cruzado se analizará un método de mediciones activas que consiste en el envío de pares de paquetes de prueba a una tasa conocida y medir a la salida del enlace los tiempos entre arribos de los paquetes de prueba.

Los paquetes de prueba se envían con un tiempo entre partidas  $t_{in}$ . Se supone en primera instancia que dos paquetes de prueba arriban al enlace en el mismo período de ocupación de la cola del enlace, como se muestra en la figura 8.1.

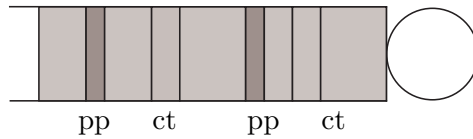


Figura 8.1: Paquetes de prueba (pp) y tráfico cruzado (ct) en la cola durante el mismo período de ocupación.

A la salida del enlace se miden los tiempos entre arribos  $t_{out}^i$  entre los paquetes  $i$  y  $i+1$ . Estos tiempos son iguales a  $\frac{X_i}{C} + \frac{K}{C}$ , donde  $X_i$  es la cantidad de tráfico cruzado que arribó al enlace entre los paquetes de prueba  $i$  e  $(i+1)$ .  $K$  es el tamaño de los paquetes de prueba y  $C$  es la capacidad del enlace.  $C$  y  $K$  son constantes y  $K$  debería ser lo más chico posible para generar la menor distorsión posible al monitorear la red. El tiempo entre arribos es entonces proporcional al volumen de tráfico cruzado a menos de una constante.

Se considera ahora el caso en que el paquete de prueba  $i+1$  y el  $i$  arriban a la cola en diferentes períodos de ocupación de la misma. En este caso como el tráfico cruzado se infiere de los valores  $t_{out}^i$ , se concluirá que existe un volumen de tráfico cruzado mayor al que realmente existe.

Baccelli et al. [93] presentan un riguroso análisis probabilístico de los métodos de mediciones activas basados en pares de paquetes de prueba para estimar tráfico cruzado. En particular analizan la identificabilidad del sistema y muestran que en el caso general diferentes tipos de tráfico cruzado pueden dar lugar a la misma secuencia de retardos observados en los tiempos de los paquetes de prueba y que no siempre es posible estimar la distribución de cualquier característica del tráfico cruzado usando paquetes de prueba.

Sin embargo, es importante resaltar que el objetivo de este trabajo no es tener un estimador preciso del tráfico cruzado, sino tener un estimador preciso de la variable  $Y$ . Por lo tanto, el interés

se centra en buscar un buen indicador del tráfico cruzado que permita distinguir entre los diferentes tipos de este tráfico que utilizan el enlace en cuestión. Obviamente cuanto mejor sea este indicador, se podrá distinguir con él mejor entre los tipos de tráfico cruzado y por lo tanto se tendrá una estimación más precisa de  $Y$

Se denominará  $X$  a la variable que permite distinguir entre los diferentes tipos de tráfico cruzado que son observables en el enlace. Con esta variable se planteará un problema de regresión. A partir de las parejas  $(X, Y)$  medidas se buscará estimar la función que las relaciona.

A los efectos de estimar  $\phi$  se asumirá que verifica un modelo de regresión, es decir que se verifica  $Y = \phi(X) + \varepsilon$ , donde  $X$  es una variable que caracteriza el estado de la red y  $\varepsilon$  es una variable aleatoria centrada correspondiente al error.

Se divide el proceso de estimación en dos fases. En la primera fase denominada fase de entrenamiento, se envía una secuencia de paquetes de prueba de tamaño fijo  $K$  separados un tiempo fijo  $t_{in}$ . Luego de estos paquetes de prueba, se envía una secuencia de video corta. Este procedimiento es repetido periódicamente como se muestra en la figura 8.2.

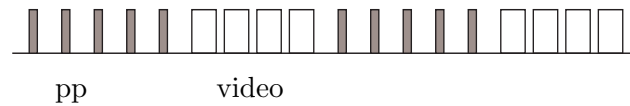


Figura 8.2: Secuencia de entrenamiento: paquetes de prueba y video.

Con la información de los tiempos entre arribos de los paquetes de prueba, se construirá la variable  $X$  que caracteriza el estado de la red observado durante cada prueba. Específicamente, una estimación de dicho estado, se extraerá de los tiempos de arribo  $t_{out}$  entre paquetes de prueba consecutivos, ya que esta información está correlacionada con el estado de la red.

En este trabajo, se estudiarán diferentes casos de estimación. En estos casos, se usarán diferentes variables  $X$  para estimar el estado de la red a partir de la información de tiempos entre arribos de los paquetes de pruebas.

Usando esta variable  $X$  y midiendo la variable de interés  $Y$  sobre el tráfico de video, se estimará  $\phi$  por una función  $\phi_n$ .

En la segunda fase, denominada fase de monitoreo, se envían solamente paquetes de prueba a los efectos de caracterizar el estado de la red  $X$ . Usando la función  $\phi_n$  estimada en la primera fase se estima el parámetro de interés  $\hat{Y}$  utilizando la estimación de la función aprendida en la fase de entrenamiento  $\hat{Y} = \phi_n(X)$ , y sin la necesidad de enviar el video.

Para cada secuencia de paquetes de prueba y video  $j$  se tiene una pareja  $(X_j, Y_j)$ , y el problema es estimar la función  $\phi : \mathcal{D} \rightarrow \mathbb{R}$  de estas observaciones, donde  $\mathcal{D}$  es el espacio al que pertenece la variable  $X$ . Para este fin se utilizará como método de regresión Nadaraya-Watson en este capítulo y Support Vector Machines en el siguiente.

El otro aspecto a tener en cuenta es que el tráfico cruzado es un proceso dependiente y no estacionario. Este punto ha sido analizado por diversos autores en los últimos años. Zhang et al. [162,

163] muestran que muchos procesos en Internet (como las pérdidas de paquetes por ejemplo) pueden ser bien modeladas como una variable aleatoria i.i.d. dentro de una “change free region”, donde la estacionariedad puede ser asumida. Ellos describen el comportamiento de la red como una serie de intervalos estacionarios a tramos. Existen diversos trabajos donde se analiza el comportamiento del tráfico en Internet. Es importante notar que las características del tráfico en Internet cambian año a año por el crecimiento de los usuarios y el tipo de aplicaciones más utilizadas en la red. Varios trabajos realizados en diferentes años, analizan que el tráfico en Internet se comporta de manera diferente en diferentes escalas de tiempo y también dependiendo del lugar de la red donde se mida. En el trabajo de Karagiannis et al. [76] se observa por ejemplo que en el corazón de la red a escalas de tiempo muy pequeñas (a escala de paquetes en enlaces de Gbps) el tráfico se distribuye como un proceso de Poisson, mientras que se observa tráfico no estacionario y con dependencias largas en escalas de tiempo mayores. Los autores proponen también un modelo de Poisson dependiente del tiempo que permite representar las características de estas diferentes escalas de tiempo. En un trabajo reciente de Noirie et al. [110], se analizan también las variaciones que presenta el tráfico de Internet cuando es observado a diferentes escalas de tiempo. La no-estacionariedad tiene diferentes causas y escalas de tiempos y la escala de tiempos estacionaria puede ser diferente para diferentes caminos en la red. Por estas consideraciones, es importante el resultado demostrado en el teorema 8.2 ya que permitirá en la siguiente sección aplicar métodos de regresión sobre variables funcionales no estacionarias y saber que el estimador es consistente.

### Consideraciones sobre las hipótesis realizadas

En el análisis anterior se consideró el caso de un enlace solamente. Se realizarán consideraciones ahora cuando se tiene un camino con varios enlaces. En primer lugar se debe resaltar que en muchos casos de interés el análisis de un camino con varios enlaces puede ser reducido al análisis de un solo enlace. Por ejemplo este es el caso cuando el servicio se ofrece a través de un servidor localizado en el backbone de un ISP (un servidor de video bajo demanda por ejemplo) y el usuario accede a ese backbone a través de un enlace celular o de una conexión ADSL. En estos casos normalmente el cuello de botella está localizado en el acceso y este enlace es el único que importa ya que el backbone está normalmente sobredimensionado. En el caso donde los paquetes realmente deben esperar en más de una cola, cada una de ellas modificará el tiempo entre paquetes que se utiliza como estimación del estado de la red. Esto significa que la variable que estima el estado de la red tendrá acumulada la influencia de todas las colas. En el próximo capítulo se estudiará este punto con más detalle para un estimador particular del estado de la red. De todas formas aún en este caso, el método de estimación mediante aprendizaje estadístico estimará correctamente la variable  $Y$  si es posible distinguir con la variable  $X$  los diferentes valores del estado de la red en el camino en cuestión.

Otra hipótesis realizada es que en el camino que se está analizando, las capacidades de los enlaces y el tamaño de los buffers son fijos. Esta hipótesis si se considera un camino fijo parece razonable ya que no son valores que se modifiquen con frecuencia. Sin embargo, la ruta entre dos puntos de la red puede cambiar. Eso dependerá de si el camino en cuestión atraviesa un único dominio o varios dominios. Si bien estos cambios sobre un camino dado no se suceden todos los días, si podría suceder que durante un período de monitoreo en cierto momento la ruta cambie. En cualquier

caso, este problema puede ser solucionado por la aplicación que realiza el monitoreo de la red. Es posible por ejemplo verificar periódicamente la ruta entre dos puntos utilizando una aplicación como traceroute. Si una nueva ruta se detecta se pueden dar diferentes situaciones. Si el sistema de monitoreo tiene ya aprendida de antes información para el nuevo camino, esta información puede ser usada para la estimación. Si el sistema no tiene información aprendida para esta nueva ruta, es necesario comenzar una nueva fase de aprendizaje. Otra alternativa es definir un umbral para las distancias entre las variables  $X$  que estiman el estado de la red. Cuando la aplicación detecta que la distancia entre la muestra actual  $X_j$  y todas la muestras de aprendizaje es mayor que dicho umbral se puede disparar el inicio de una nueva fase de entrenamiento. Esta técnica de los umbrales sobre las distancias permite también detectar otras variaciones importantes del estado de la red y no sólo un cambio de ruta. Finalmente es importante resaltar que cuando el cuello de botella se encuentra en el enlace de acceso, y el backbone está sobredimensionado es posible que un cambio en la ruta no afecte las mediciones.

Otra cuestión que debe ser analizada es la hipótesis de que el sistema es entrenado para un único tipo de video (se asumió que  $V_t$  es una secuencia fija). Esto no es una limitante muy importante ya que la QoS de secuencias de video dependen de un conjunto acotado de características del mismo como por ejemplo: el codec usado, la tasa de bits y el nivel de movimiento del video. Por lo tanto, es posible entrenar el sistema con diferentes tipos de videos de acuerdo al conjunto de valores de los parámetros anteriores que se utilicen en el servicio en cuestión (utilizando un conjunto típico de secuencias de test para cada caso). Luego, dependiendo del video específico que el usuario quiera ver, o que el operador quiera evaluar, el sistema utilizará el conjunto correspondiente de secuencias de entrenamiento.

Por último, es necesario resaltar que en la fase de entrenamiento del sistema propuesto, periódicamente se envía una secuencia de paquetes de prueba y luego, separado en el tiempo, una secuencia de video. Dependiendo de la aplicación específica, se puede utilizar también el envío conjunto de los paquetes de prueba y la secuencia de video en la fase de entrenamiento. En el caso de enviarlos por separado, se predice la QoS que recibiría un video si hubiera sido enviado por el camino. En el caso de enviarlos juntos se predice la calidad de servicio de un video que está siendo enviado por el camino. En el análisis anterior, se propuso el envío por separado de los paquetes de prueba y la secuencia de video porque para aplicaciones de control de admisión a una red, que es un caso de interés para este análisis, esta aproximación es más adecuada. Sin embargo, ambos casos pueden ser analizados con la metodología propuesta.

#### **8.4.2. Estimación de la QoS a partir de la distribución de tiempos entre paquetes de prueba**

En este caso la variable  $X$  será la distribución empírica de los tiempos entre arribos de paquetes de prueba consecutivos. Para evaluar la aproximación de la estimación mediante el proceso de regresión, se presentan a continuación los resultados para datos simulados con el simulador de redes Ns-2 [100]. Se simula el tráfico en un enlace según un proceso ON-OFF. El proceso ON-OFF considerado tiene media variable en dos rangos diferentes, un rango correspondiente a un régimen de poco tráfico y otro correspondiente a un régimen de mucho tráfico. Los tiempos ON y OFF corresponden a variables exponenciales independientes. Luego se quiere simular el comportamiento



no estacionario del tráfico, mediante un variable se selecciona periódicamente entre el régimen de poco tráfico y el de mucho tráfico. Para cada tipo de tráfico se sortea la media de las exponenciales y la media de tráfico dentro de uno de dos rangos. Para el primer rango la tasa de bits del tráfico tiene media entre 150 Mb/s y 450 Mb/s y tiempo promedio en estado ON y OFF variando entre 100 y 300 ms. En el segundo rango la tasa media es entre 600 Mb/s y 900 Mb/s y el tiempo medio en el estado ON y OFF varía entre 200 500 ms. Con la media de tráfico sorteada y los tiempos ON y OFF se determina el tiempo entre paquetes en el período ON de modo de tener la media de tráfico sorteada para la realización correspondiente del proceso ON-OFF.

Asumiendo este tráfico en el enlace se desarrolla el procedimiento de medición descrito antes. Se realizan pruebas que consisten en envío de paquetes periódicos livianos para los que se calcula el tiempo entre llegadas y luego de cada ráfaga de paquetes de prueba se envía una secuencia corta que simula el video, con un modelo ON-OFF, para la cual se calcula el retardo medio.

Para el tráfico simulado se tienen 360 observaciones  $(X, Y)$ . Se separa la muestra en 2,  $(X^1, Y^1)$  con los valores de las observaciones para  $i = 1, \dots, 120$  (muestra de validación) y  $(X^2, Y^2)$  con los valores de las observaciones para  $i = 121, \dots, 360$  (muestra de entrenamiento). Se calcula el estimador  $\phi_n(X_i^1)$ , para cada  $i = 1, \dots, 120$ , a partir de las 240 observaciones de la segunda muestra. El núcleo es  $K(x) = (x^2 - 1)^2 1_{\{|x| < 1\}}$ .

Antes de utilizar como variable  $X$  la distribución empírica, la primera aproximación que se utilizó fue tratar de tomar como variable  $X$  el valor medio o la desviación estándar del tiempo entre arribos de los paquetes de prueba. En la figura 8.3 se observa el valor medio y la desviación estándar de los tiempos entre arribos de los paquetes de prueba y el valor de la variable  $Y$  (retardo del video) para las secuencias correspondientes. Por simple observación se ve puede apreciar lo poco correlacionado que están estos datos. Esto se puede apreciar mejor cuando se intenta realizar una regresión tomando la media o la varianza como variable  $X$ . Los resultados se muestran en la figura 8.4.

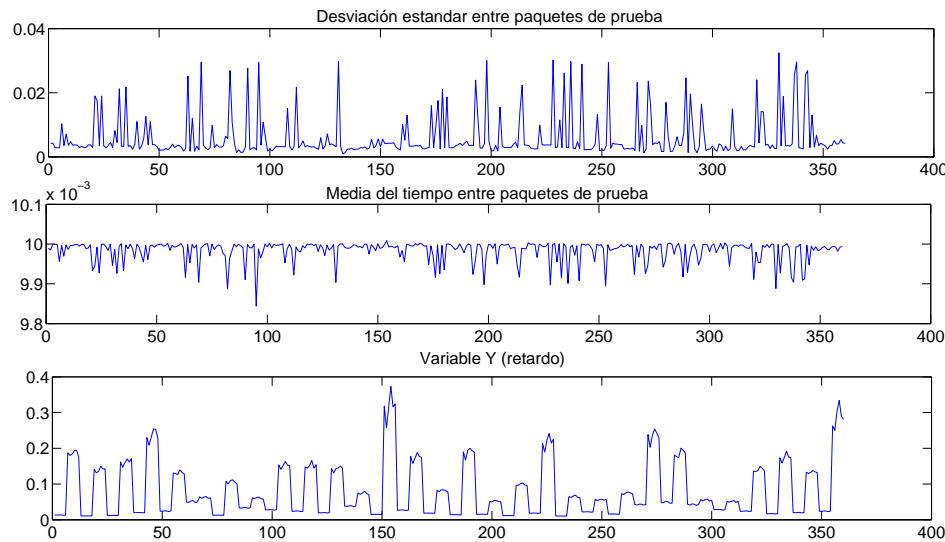


Figura 8.3: Y

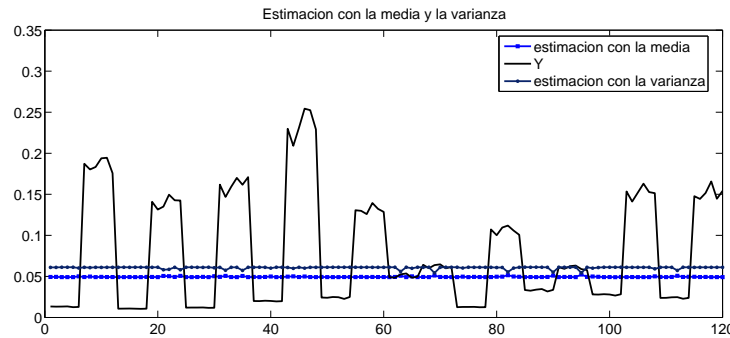


Figura 8.4:  $Y$ , valor medido y su estimación usando la media y la varianza

Como se puede apreciar en dicha figura, este camino no parece aportar información sobre la variable a estimar  $Y$ . Por este motivo se decidió ver si realmente existía información en estos tiempos mirando la distribución empírica de los mismos.

Teniendo en cuenta lo anterior se consideran datos  $(X, Y)$  donde  $X$  es la distribución empírica del tiempo entre paquetes e  $Y$  es algún parámetro de calidad de servicio medido sobre la secuencia de video. Se asume que los datos están en las hipótesis de los resultados presentados en la sección 8.3, es decir que  $X$  es funcional y es una mezcla de un proceso estacionario débilmente dependiente y de otro que puede ser no estacionario y dependiente. Teniendo en cuenta el modelo utilizado para probar la convergencia del estimador de Nadaraya-Watson en el caso no estacionario, se está asumiendo que  $X$  representa el estado de la red, y es de la forma  $X = \varphi(\xi, Z)$ , donde el proceso  $\xi$  es estacionario y débilmente dependiente pero  $Z$  no lo es. Se modela mediante  $Z$  la selección entre diferentes regímenes de tráfico, donde esto depende por ejemplo de la hora, del día, etc.. Por otra parte para cada tipo de tráfico, es decir condicionado a  $Z$ , se tendrá un comportamiento estacionario determinado por  $\xi$ .

El modelo puede interpretarse de la siguiente manera. La variable  $\xi$  toma valores en un conjunto de funciones de distribución, que son las funciones de distribución del tiempo entre paquetes. Éstas varían según la media del tráfico, donde existen dos rangos para la media, correspondientes a mayor y menor tráfico presente en la red. La variable  $Z$  elige entre los dos grupos determinados por mayor o menor media, y luego dentro de cada grupo se sortea una función de distribución, correspondiente al sorteo de la media de tráfico de manera uniforme en el rango seleccionado.

Es decir que cada realización del proceso ON-OFF del tráfico tiene asociada una función de distribución empírica del tiempo entre paquetes de prueba y un número correspondiente al retardo medio medido sobre la secuencia de video simulado.

Como seminorma en el espacio de las distribuciones empíricas se utilizó la norma  $L^1$ . De esta forma se estima la “distancia” entre dos variables  $X_i$ .

En las figuras 8.5, 8.6 se presentan la estimaciones para los valores de la muestra de validación y el error relativo. Como puede observarse se obtiene una buena aproximación. El error relativo es menor o del orden del 10% en casi todos los puntos excepto en una zona donde el error está entre el 15 y el 25%.

El error en esta zona tiene dos componentes (ambas relacionadas). Por un lado hay pocos puntos

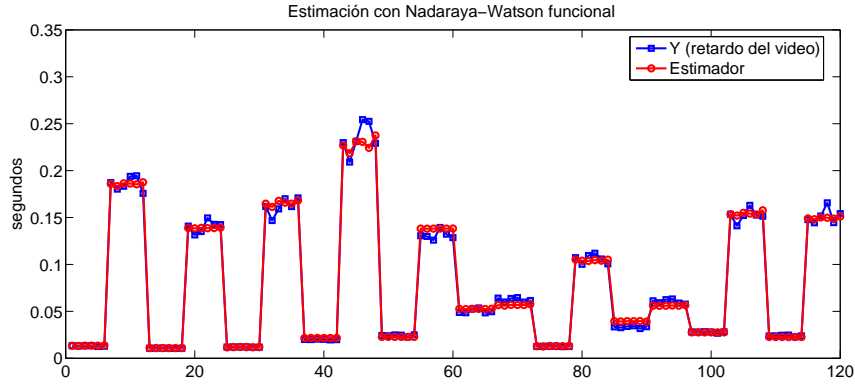


Figura 8.5: Retardo estimado para datos simulados.

en la muestra de aprendizaje en ese rango de valores lo que lleva a aproximaciones menos aproximadas. Por otra parte, hay que tener en cuenta el valor de la ventana del núcleo utilizada. El valor de la ventana del estimador de NW depende del punto a estimar. Esto para muchas aplicaciones es una restricción fuerte, porque se debe para cada punto a estimar calcular un valor de la ventana, lo cual presenta dificultades tanto en la técnica de cálculo cómo en el costo computacional. Por eso se busca en muchos casos, un valor de la ventana de forma que estime de manera “razonable” el conjunto de puntos de la muestra. El procedimiento seguido en el ejemplo anterior para calcular la ventana es el siguiente. Para la muestra de entrenamiento,  $\{(X_i, Y_i) : 1 \leq i \leq n\}$  se considera la estimación del valor  $Y_i$  definido por  $\hat{Y}_i = \phi_{n,i}(X_i)$  donde  $\phi_{n,i}$  es obtenido de la muestra de entrenamiento sin considerar el punto  $(X_i, Y_i)$ . Luego se calcula:

$$\sigma^2(h_n) = \frac{1}{n} \sum_{i=1}^n (\phi_{n,i}(X_i) - Y_i)^2$$

para un conjunto finito de ventanas  $H$  y se selecciona aquella  $h_n^*$  tal que:

$$\sigma^2(h_n^*) = \min\{\sigma^2(h_n) : h_n \in H\}.$$

El problema que tiene este procedimiento como se dijo es que busca una ventana que en promedio estime bien. Sin embargo, puede haber puntos donde exista un error de estimación apreciable. Este procedimiento se puede mejorar por ejemplo partiendo la muestra de entrenamiento en diferentes “zonas” y seleccionando un  $h_n$  para cada zona, buscando la combinación de ventanas que minimice el error cuadrático medio.

Para probar este método para estimar la ventana y ver cuánto mejora la estimación anterior, se dividieron los datos en cuatro “zonas” dependiendo del valor de  $Y_i$  en la muestra de entrenamiento ( $0 < Y_i < 50ms$ ,  $50 < Y_i < 100ms$ ,  $100 < Y_i < 200ms$ ,  $200ms < Y_i$ ). Se calculó una ventana para cada zona. Luego cuando se tiene un nuevo punto  $x$  a estimar, se busca el punto  $X_i$  de la muestra de entrenamiento que esté a menor distancia y se utiliza el valor de la ventana de ese punto.

Como se puede ver en las figuras 8.7 y 8.8 el error máximo disminuyó de 25 a 20 % y en la zona de mayor error se pasó de errores entre 15 y 25 % a errores entre 10 y 20 %. Si bien, en esta figura

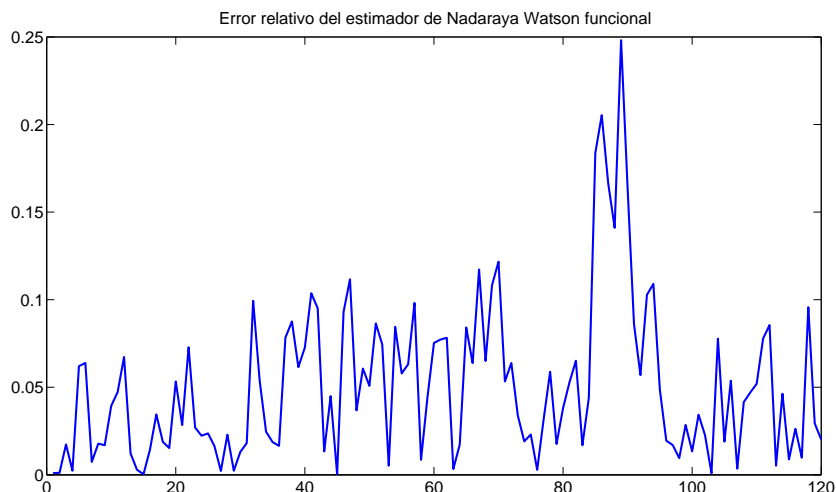


Figura 8.6: Error relativo para datos simulados.

se muestran sólo los puntos donde el error de estimación era mayor, el error cuadrático medio de toda la muestra de test también disminuyó.

## 8.5. Conclusiones

En este capítulo en primer lugar se introduce el estimador de Nadaraya-Watson. Utilizando este método de regresión y la distribución de los tiempos entre paquetes de pruebas como estimador del estado del camino, se propone un método para estimar la calidad de servicio que recibiría una aplicación. Para poder aplicar Nadaraya-Watson al problema de estimación de la QoS en Internet se extiende un resultado de Ferraty et al. sobre la convergencia del estimador funcional de Nadaraya-Watson para el caso no estacionario.

La metodología anterior presenta algunas debilidades si se piensa en el objetivo de desarrollar un sistema de monitoreo de la QoS. La primera es, la estimación de la ventana del núcleo. Este parámetro es bastante crítico y el grado de aproximación del estimador depende fuertemente de este parámetro. Se puede mejorar la estimación por diferentes mecanismos como haciendo bootstrap en los datos de la muestra de entrenamiento, pero mejorar la estimación para cada punto implica mayores tiempos de entrenamiento y cómputo.

La segunda debilidad es el costo computacional por el uso de datos funcionales. Si bien con datos funcionales se logra una buena aproximación, esto en la práctica tiene como desventaja que es necesario almacenar una cantidad de información importante para cada punto. Para sistemas de monitoreo en línea y donde se pueden requerir registros históricos largos esto se puede transformar en un problema. También es cierto que puede haber zonas de la distribución empírica que no aporten información relevante o que incluso tiendan a disminuir la precisión del estimador y este método no lo detecta.

La tercera desventaja es que al tomar como variable  $X$  la distribución empírica de los tiempos

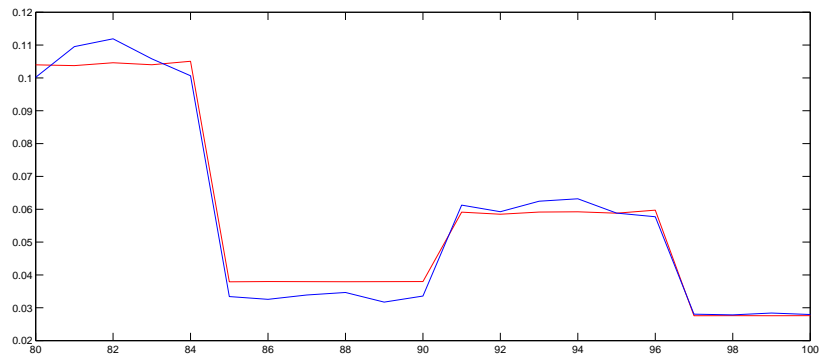


Figura 8.7: Retardo estimado para datos simulados.

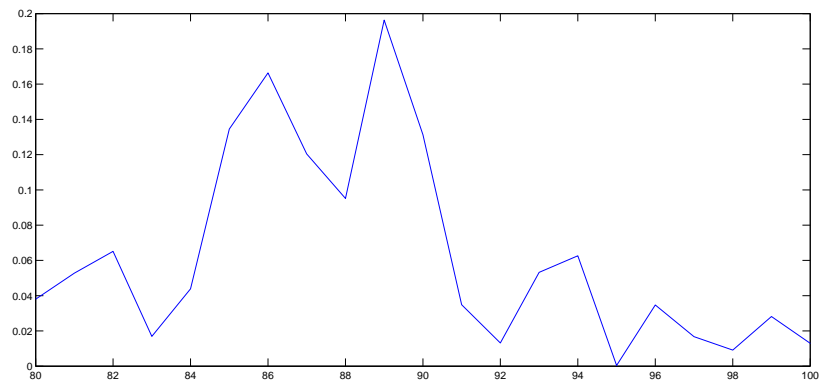


Figura 8.8: Error relativo para datos simulados.

entre paquetes se está perdiendo información de correlación del proceso de tráfico cruzado. En realidad, no se pierde totalmente esta correlación porque al medirlo en la cola de un enlace algo de su correlación está implícita en la medida. Esto se debe a que el proceso tuvo un volumen de tráfico acumulado en el tiempo capaz de superar la capacidad del enlace como para formar cola y por eso los paquetes de prueba fueron capaces de detectarlos. Sin embargo, si se pierde como es la correlación de este proceso en la propia cola, lo que tiene que ver por ejemplo con la ocupación máxima y media que alcanza la cola, que son indicadores de cómo es el tráfico cruzado.

Por último, el método de Nadaraya-Watson tiene un inconveniente para aplicaciones en línea y es que para cada nuevo punto a estimar se debe calcular la distancia de este punto con todos los puntos de la muestra de entrenamiento y eso puede tener un costo de almacenamiento y computacional alto, y más aún si esa distancia es sobre un modelo funcional.

Por estas razones se buscará en el próximo capítulo proponer algunas técnicas que mejoren las debilidades mencionadas.

## 8.6. Apéndice: Demostraciones

### 8.6.1. Prueba del lema 8.1

*Demostración.*

$$E(f_n(x)) = \frac{1}{n\psi(h_n)} \sum_{i=1}^n E(K_n(X_i))$$

Como  $\xi$  y  $Z$  son independientes se cumple

$$E(K_n(X_i)) = E\{E(K_n(X_i)|Z_i)\} = \sum_{k=1}^m E\{K_n(\varphi(\xi_i, z_k))\}P(Z_i = z_k)$$

Como para cada  $k$   $(\varphi(\xi_i, z_k))_{i \geq 1}$  es una sucesión estacionaria se tiene que

$$E(f_n(x)) = \sum_{k=1}^m \left( \frac{1}{\psi(h_n)} E(K_n(\varphi(\xi_1, z_k))) \frac{1}{n} \sum_{i=1}^n P(Z_i = z_k) \right).$$

La hipótesis 8.3 implica que la densidad de  $\|\varphi(\xi_1, z_k) - x\|$  es la función  $u \mapsto c_k(x)\psi'_k(u, x)$  y entonces

$$E[K_n(\varphi(\xi_1, z_k))] = h_n c_k(x) \int_0^1 K(u)\psi'_k(uh_n, x) du$$

Luego usando las hipótesis 8.3, 8.4, 8.8 se tiene que

$$\lim_{n \rightarrow \infty} \frac{1}{\psi(h_n)} E[K_n(\varphi(\xi_1, z_k))] = d_k(x)c_k(x)\mathbf{1}_{\{k \in \Delta\}},$$

y usando la hipótesis 8.5

$$\begin{aligned} \lim_{n \rightarrow \infty} E(f_n(x)) &= \lim_{n \rightarrow \infty} \sum_{k=1}^m \left( \frac{1}{\psi(h_n)} E(K_n(\varphi(\xi_1, z_k))) \frac{1}{n} \sum_{i=1}^n P(Z_i = z_k) \right) \\ &= \sum_{k \in \Delta} p_k d_k(x) c_k(x) \end{aligned}$$

Análogamente se tiene que

$$E(g_n(x)) = \sum_{k=1}^m \left( \frac{1}{\psi(h_n)} E(\phi(\varphi(\xi_1, z_k)) K_n(\varphi(\xi_1, z_k))) \frac{1}{n} \sum_{i=1}^n P(Z_i = z_k) \right).$$

Entonces

$$E(g_n(x)) = \phi(x)E(f_n(x)) + R_n,$$

donde

$$\begin{aligned} R_n &= \sum_{k=1}^m \left( \frac{1}{\psi(h_n)} E[\{\phi(\varphi(\xi_1, z_k)) - \phi(x)\} K_n(\varphi(\xi_1, z_k))] \frac{1}{n} \sum_{i=1}^n P(Z_i = z_k) \right) \\ &\leq \sup_{u: \|x-u\| \leq h_n} |\phi(u) - \phi(x)| E(f_n(x)) \end{aligned}$$

y el resultado  $\lim_{n \rightarrow \infty} E(g_n(x)) = \phi(x)f(x)$  se obtiene a partir de la continuidad de  $\phi$  y de  $\lim_{n \rightarrow \infty} E(f_n(x)) = f(x)$ .  $\square$

### 8.6.2. Prueba del Teorema 8.2

*Demostración.* Como  $\phi_n(x) = \frac{g_n(x)}{f_n(x)}$  basta probar que  $f_n(x)$  converge casi seguramente a  $f(x)$  y que  $g_n(x)$  converge casi seguramente a  $\phi(x)f(x)$ . Se tiene que

$$f_n(x) - f(x) = f_n(x) - E(f_n(x)) + E(f_n(x)) - f(x)$$

$$g_n(x) - \phi(x)f(x) = g_n(x) - E(g_n(x)) + E(g_n(x)) - \phi(x)f(x)$$

y de la proposición 8.1

$$\lim_{n \rightarrow \infty} E(f_n(x)) - f(x) = 0,$$

$$\lim_{n \rightarrow \infty} E(g_n(x)) - f(x)\phi(x) = 0,$$

entonces basta probar que  $f_n(x) - E(f_n(x))$  y  $g_n(x) - E(g_n(x))$  convergen a cero casi seguramente. Para probar que  $f_n - E(f_n(x))$  converge a cero casi seguramente basta probar la convergencia completa, es decir que para todo  $\varepsilon > 0$  la serie  $\sum P(|f_n(x) - E(f_n(x))| > \varepsilon)$  es convergente.

$$\begin{aligned} f_n(x) - E(f_n(x)) &= \frac{1}{n\psi(h_n)} \sum_{i=1}^n [K_n(X_i) - E(K_n(X_i))] \\ &= \frac{1}{\sqrt{n\psi(h_n)}} S_n \end{aligned}$$

donde

$$S_n = \frac{1}{\sqrt{n\psi(h_n)}} \sum_{i=1}^n [K_n(X_i) - E(K_n(X_i))]$$

Usando la desigualdad de Markov se tiene que

$$P(|f_n(x) - E(f_n(x))| > \varepsilon) \leq \frac{E[f_n(x) - E(f_n(x))]^4}{\varepsilon^4} = \frac{E(S_n)^4}{(n\psi(h_n))^2 \varepsilon^4}$$

Para calcular  $E(S_n^4)$  condicionamos a toda la trayectoria del proceso  $Z$  donde una trayectoria del proceso  $Z$  es la variable  $Z^\infty$  que cumple que  $Z^\infty = z^\infty$  si  $Z_i = z_{j_i}$  para todo  $i \geq 1$ . Es decir que una trayectoria es una variable aleatoria que toma valores en el espacio de sucesiones a valores en  $\{z_1, \dots, z_m\}$ .

$$\begin{aligned} E(S_n^4) &= E[E(S_n^4|Z^\infty)] \\ &= \int_T E(S_n^4|Z^\infty = z^\infty) d\mu \end{aligned}$$

siendo  $T$  el espacio de trayectorias, es decir que  $T$  es el conjunto de las sucesiones toman valores en  $\{z_1, z_2, \dots, z_m\}$ . Para probar la convergencia completa basta probar que para un conjunto de trayectorias con probabilidad 1 se cumple que  $E(S_n|Z^\infty = z^\infty) \leq a_n$  donde  $\sum \frac{a_n}{(n\psi(h_n))^2}$  converge.

Calculando para cada trayectoria

$$\begin{aligned} E(S_n^4|Z^\infty = z^\infty) &= E\left[\frac{1}{\sqrt{n}} \sum_{i=1}^n \frac{\{K_n(\varphi(\xi_i, z_{l_i})) - E[K_n(\varphi(\xi_i, z_{l_i}))]\}}{\sqrt{\psi(h_n)}}\right]^4 \\ &= E\left[\frac{1}{\sqrt{n}} \sum_{i=1}^n \tilde{X}_i^{l_i, n}\right]^4 \end{aligned}$$

con  $l_i \in \{m+1, \dots, 2m\}$ . Agrupando los sumandos según los valores de  $Z$  se tiene

$$\sum_{i=1}^n \tilde{X}_i^{l_i, n} = \sum_{k=m+1}^{2m} \sum_{i \in A_n^k} \tilde{X}_i^{k, n}$$

siendo  $A_n^k = \{i : Z_i = z_k\} \cap [1, n]$  y entonces

$$\begin{aligned} \frac{1}{\sqrt{n}} \sum_{i=1}^n \tilde{X}_i^{l_i, n} &= \sum_{k=m+1}^{2m} \frac{1}{\sqrt{n}} \sum_{i \in A_n^k} \tilde{X}_i^{k, n} \\ &= \sum_{k=m+1}^{2m} S_n(A^k, \tilde{X}^{k, n}) \end{aligned}$$



Se tiene que, aplicando la desigualdad de Holder y la hipótesis 8.9,

$$\begin{aligned}
& E \left[ \sum_{k=m+1}^{2m} S_n \left( A^k, \tilde{X}^{k,n} \right) \right]^4 = \\
& E \left[ \sum_{\substack{k_1; k_2; \dots; k_m \\ k_1 + k_2 + \dots + k_m = 4}} \frac{4!}{k_1! k_2! \dots k_m!} S_n^{k_1} \left( A^{m+1}, \tilde{X}^{m+1,n} \right) \dots S_n^{k_m} \left( A^{2m}, \tilde{X}^{2m,n} \right) \right] = \\
& \sum_{\substack{k_1; k_2; \dots; k_m \\ k_1 + k_2 + \dots + k_m = 4}} \frac{4!}{k_1! k_2! \dots k_m!} E \left[ S_n^{k_1} \left( A^{m+1}, \tilde{X}^{m+1,n} \right) \dots S_n^{k_m} \left( A^{2m}, \tilde{X}^{2m,n} \right) \right] \leq \\
& \sum_{\substack{k_1; k_2; \dots; k_m \\ k_1 + k_2 + \dots + k_m = 4}} \frac{4!}{k_1! k_2! \dots k_m!} E \left[ S_n^4 \left( A^{m+1}, \tilde{X}^{m+1,n} \right) \right]^{\frac{k_1}{4}} \dots E \left[ S_n^4 \left( A^{2m}, \tilde{X}^{2m,n} \right) \right]^{\frac{k_m}{4}} \leq \\
& \sum_{\substack{k_1; k_2; \dots; k_m \\ k_1 + k_2 + \dots + k_m = 4}} \frac{4!}{k_1! k_2! \dots k_m!} C(k_1, \dots, k_m) \left( \frac{\text{card}(A_n^{m+1})}{n} \right)^{\frac{k_1}{2}} \dots \left( \frac{\text{card}(A_n^{2m})}{n} \right)^{\frac{k_m}{2}}
\end{aligned}$$

Entonces, como  $\text{card}(A_n^k) \leq n$ , se tiene que  $E(S_n^4 | Z^\infty = z^\infty) \leq C$ . La demostración anterior, donde se probó que  $E(S_n^4 | Z^\infty = z^\infty) \leq C$  es independiente de la trayectoria. Para la convergencia completa es necesario elegir  $h_n$  de modo que la serie

$$\sum \frac{1}{(n\psi(h_n))^2}$$

sea convergente, por ejemplo  $(n\psi(h_n))^2 \geq n^\alpha$  con  $\alpha > 1$ , es decir  $\psi(h_n) \geq \frac{1}{n^{\beta/2}}$ , con  $0 < \beta < 1$  y usando la hipótesis 8.10 se tiene la convergencia completa de  $f_n(x) - E[f_n(x)]$  a cero. La convergencia completa de  $g_n(x) - E[g_n(x)]$  a cero se prueba de la misma manera, considerando las variables  $\tilde{X}^{k,n}$  con  $k \in \{1, \dots, m\}$ .  $\square$

## Capítulo 9

# Estimación de la QoS utilizando un estimador de la cola del enlace y Support Vector Machines

### 9.1. Introducción

En este capítulo se analizarán las debilidades del método de estimación propuesto en el capítulo anterior.

Una de las propuestas que se evalúan en este capítulo es utilizar un estimador proveniente de la técnica de aprendizaje estadístico denominada Support Vector Machines (SVM) [143]. Por este motivo en la sección 9.2 se hará un resumen de las ideas principales de SVM. En la sección 9.3 se propondrá un estimador del estado de la red alternativo a la función distribución empírica de los tiempos de arribo entre paquetes de prueba. En la sección 9.4 se evaluará la aplicación de este estimador al problema de regresión con Nadaraya-Watson para estimar la QoS de extremo a extremo. En la sección 9.5 se utilizará este nuevo estimador del estado de la red para estimar la QoS de servicio de extremo a extremo pero utilizando SVM en lugar de Nadaraya-Watson, y se compararán ventajas y desventajas de estas técnicas de regresión para esta aplicación.

### 9.2. Support Vector Machines

En primer lugar se estudiarán las ideas generales de SVM para el caso de clasificación ya que por un lado es el uso más extendido y por otro es más fácil en este contexto comprender la idea general. Luego se extenderán las ideas de SVM para clasificación al caso de regresión.

#### 9.2.1. SVM para clasificación

SVM es un sistema de aprendizaje supervisado. Sea  $(X_1, Y_1) \dots (X_n, Y_n)$  una muestra de aprendizaje i.i.d. con  $X_i \in \mathbb{R}^d$ ,  $Y_i \in \{-1, 1\}$ . Para clasificar los puntos de la muestra de aprendizaje, se buscará un hiperplano que los separe. Una vez aprendido este hiperplano con la muestra de

aprendizaje, dado un nuevo  $X_i$ , se analizará de que lado del hiperplano queda ubicado y en base a esta información se dirá a que clase pertenece.

Se define el hiperplano  $H = \{x : f(x) = x^t\beta + \beta_0 = 0\} \subset \mathbb{R}^d$ , donde  $\beta$  es un vector  $\mathbb{R}^d$  ortonormal al hiperplano y  $\beta_0$  un real.

Se define  $T^+ = \{X_i : Y_i = 1\}$  y  $T^- = \{X_i : Y_i = -1\}$

**Caso 1: muestra separable.** Se asumirá en primera instancia que  $T^+$  y  $T^-$  son perfectamente separables por un hiperplano.

Para un punto  $x$  del espacio  $f(x)$  es la distancia con signo al hiperplano  $H$ . Un punto  $(X, Y)$  estará bien clasificado si  $Yf(X) > 0$ .

La idea de SVM es encontrar el hiperplano que deje el mayor margen de separación con los datos de la muestra de aprendizaje.

Sea  $C$  la distancia del punto de la muestra de aprendizaje más cercano al hiperplano.

Lo que se busca es maximizar:

$$\begin{aligned} & \underset{\beta, \beta_0}{\text{máx}} C \\ \text{sujeto a : } & Y_i(\langle X_i, \beta \rangle + \beta_0) \geq C \\ & \|\beta\| = 1, \beta_0 \in \mathbb{R} \end{aligned}$$

Haciendo un cambio de variable  $\tilde{\beta} = \frac{\beta}{C}$  este problema fácilmente se transforma en el siguiente problema de optimización convexa:

$$\begin{aligned} & \underset{\beta, \beta_0}{\text{mín}} \frac{1}{2} \|\tilde{\beta}\|^2 \\ \text{sujeto a : } & Y_i(\langle X_i, \tilde{\beta} \rangle + \tilde{\beta}_0) \geq 1, \forall i = 1 \dots n \end{aligned} \tag{9.1}$$

Planteando entonces el Lagrangiano y las condiciones de Khun-Tucker (KKT) se formula el problema dual en función de los multiplicadores de Lagrange  $\alpha_i$ :

$$\begin{aligned} \text{máx}_{\alpha_i} \mathcal{L} = & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j Y_i Y_j \langle X_i, X_j \rangle \\ & \alpha > 0 \end{aligned}$$

Resolviendo este sistema se obtienen los multiplicadores de Lagrange (observar que hay uno por cada muestra de entrenamiento)  $\alpha^0 = (\alpha_1^0, \dots, \alpha_n^0)$ , de donde se puede calcular  $\beta^* = \sum_{i=1}^n \alpha_i^0 Y_i X_i$  y  $\beta_0^*$  se obtiene de de las condiciones KKT:

$$Y_i(\langle \beta^*, X_i \rangle + \beta_0^*) - 1 = 0$$

Observar que

$$\alpha_i(Y_i(\langle \beta^*, X_i \rangle + \beta_0^*) - 1) = 0, \forall i$$

Los puntos que están en la frontera de la restricción (los que están a distancia mínima de  $H$ ), son los que verifican que:  $Y_i(\langle \beta^*, X_i \rangle + \beta_0^*) - 1 = 0$  y por tanto son los que tienen multiplicadores de Lagrange  $\alpha_i > 0$ . Estos puntos se llaman vectores soporte. Los demás puntos tienen  $\alpha_i = 0$  y no intervienen en determinar  $\beta^*$  y  $\beta_0^*$ .

La regla de decisión será  $signo(\langle \beta^*, X \rangle + \beta_0^*)$

Se debe observar que para el cálculo de los multiplicadores de Lagrange ( $\alpha$ ), los vectores de entrenamiento  $X_i$  sólo aparecen a través de su producto interno. Esta observación será relevante luego para el caso no lineal.

### Caso 2: Caso no separable

En este caso van a existir puntos que quedarán mal clasificados y el problema como fue formulado antes no será resoluble ya que para cualquier  $C$  habrá puntos que no satisfacen la restricción. Para encontrar soluciones factibles en este caso, se debe relajar la restricción y esto se hará a través del agregado de un conjunto de variables auxiliares:  $\zeta_i$  y las restricciones del problema 9.1 quedan:

$$\begin{aligned} Y_i(\langle X_i, \beta \rangle + \tilde{\beta}_0) &\geq 1 - \zeta_i, \quad \forall i = 1 \dots n \\ \zeta_i &> 0, \quad \forall i = 1 \dots n \end{aligned} \quad (9.2)$$

Para que ocurra un error, el correspondiente  $\zeta_i$  debe ser mayor que 1 por lo que la suma de estas variables es una cota superior al número de errores de entrenamiento. La idea será entonces agregar el costo de clasificar mal a la función de costo de la optimización anterior. De esta forma el problema de optimización se modifica a:

$$\begin{aligned} \min_{\beta, \beta_0, \zeta_i} \quad & \frac{1}{2} \|\tilde{\beta}\|^2 + \gamma \sum_{i=1}^n \zeta_i \\ \text{suje}to \quad & a: \\ Y_i(\langle X_i, \tilde{\beta} \rangle + \tilde{\beta}_0) &\geq 1 - \zeta_i, \quad \forall i = 1 \dots n \\ \zeta_i &> 0, \quad \forall i = 1 \dots n \end{aligned} \quad (9.3)$$

donde  $\gamma$  es un parámetro que mide cuanto se penaliza a los errores de entrenamiento.

En este caso la resolución es similar al caso anterior. Es decir, se plantea el Lagrangiano y las condiciones KKT y eso lleva al problema dual

$$\begin{aligned} \max_{\alpha_i} \mathcal{L} = \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j Y_i Y_j \langle X_i, X_j \rangle \\ \text{suje}to \quad & a: \\ 0 &\leq \alpha_i \leq \gamma \\ \sum_{i=1}^n \alpha_i Y_i &= 0 \end{aligned}$$

de donde se obtiene al igual que en el caso anterior :

$$\beta^* = \sum_{i=1}^n \alpha_i^0 Y_i X_i$$

Al igual que antes, en realidad se debe sumar sólo sobre los vectores soporte ( $\alpha_i > 0$ ).

La diferencia ahora es que los  $\alpha_i$  están acotados por  $\gamma$  y los puntos con  $\alpha_i = \gamma$  son puntos de la muestra de entrenamiento mal clasificados (se puede ver de las condiciones KKT) y son también vectores soporte.

### Caso 3: SVM no lineal

Si no hay separación lineal entre los datos, la idea es mapear los vectores de  $\mathbb{R}^d$  a un espacio de dimensión mayor donde los datos puedan ser separados por un hiperplano.

Se busca una función  $\phi : \mathbb{R}^d \rightarrow \mathcal{F}$  donde  $\mathcal{F}$  se denomina espacio de características. Lo relevante en este proceso es que no se necesitará conocer explícitamente ni el espacio  $\mathcal{F}$  ni la función de mapeo  $\phi$ . Esto se debe a la observación que se realizó antes donde se apreciaba que la función que permite clasificar un punto  $X$  :

$$f(X) = \text{signo}\left(\sum_{i=1}^n \alpha_i^0 Y_i \langle X_i, X \rangle + \beta_0^*\right)$$

sólo depende de los puntos del espacio a través de su producto interno. Por lo tanto si en el nuevo espacio se conoce la función que determina el producto interno de dos vectores es posible clasificar linealmente en ese espacio con:

$$f(X) = \text{signo}\left(\sum_{i=1}^n \alpha_i^0 Y_i \langle \phi(X_i), \phi(X) \rangle + \beta_0^*\right)$$

Se llamará Kernel a la función que retorna el producto interno de dos vectores en el espacio  $\mathcal{F}$ :

$$K(X, X_1) = \langle \phi(X), \phi(X_1) \rangle$$

Un Kernel será una función de  $\mathbb{S}_x \times \mathbb{S}_x \rightarrow \mathbb{R}$  que verifica:

- Es simétrico:  $K(X, X_1) = K(X_1, X)$
- Es definido positivo:  $\sum_{i,j=1}^m a_i a_j K(X_i, X_j) \geq 0$ ,  $\forall m \in \mathbb{N}, a_i \in \mathbb{R}, X_i \in \mathbb{S}_x$

Por lo tanto si se conoce el Kernel no interesa conocer la transformación explícitamente.

Algunos ejemplos de Kernel son:

$$K(X, X_1) = e^{\frac{1}{c}\|X - X_1\|^2} \tag{9.4}$$

$$K(X, X_1) = \tanh(C_1 \vec{X} \vec{X}_1 + C_2) \tag{9.5}$$

El principal problema es si dada una función  $K(X, X_1)$  existe un espacio  $\mathcal{F}$  y una función  $\phi$  asociadas.

El principal resultado para responder a esta pregunta es conocido como condición de Mercer que establece que:

Si  $K(s, t)$  es definido positivo, continuo y  $K(s, t) \in L^2(S_x \times S_x)$ , entonces existe  $\{\phi_i\}_{i \in \mathbb{N}}$  ortonormal y  $\{\lambda_i\}$  valores propios no negativos tales que:

$$K(s, t) = \sum_{i=1}^{\infty} \lambda_i \phi_i(s) \phi_i(t)$$

Como corolario de este problema se puede ver que la función  $\phi : \mathbb{S}_x \rightarrow \mathbb{F}$  tal que  $\phi(x) = (\sqrt{\lambda_k} \phi_k(x))_{k \in \mathbb{N}}$  es el mapeo al espacio de características correspondiente al Kernel.

### 9.2.2. SVM en regresión

Una vez que el problema de clasificación fue analizado, el problema de regresión es más simple de explicar.

Si se comienza por el caso lineal, la idea será buscar un hiperplano que pase por los puntos de la muestra de entrenamiento. En realidad se busca un hiperplano donde todos los puntos de entrenamiento caigan dentro de una banda a distancia  $\pm \varepsilon$  del hiperplano. Entonces, el problema de optimización se reduce a:

$$\min_{\beta, \beta_0} \frac{1}{2} \|\tilde{\beta}\|^2 \quad (9.6)$$

*sujeito a :*

$$\begin{aligned} Y_i - \langle X_i, \tilde{\beta} \rangle + \beta_0 &\leq \varepsilon, \quad \forall i = 1 \dots n \\ \langle X_i, \tilde{\beta} \rangle + \beta_0 - Y_i &\leq \varepsilon, \quad \forall i = 1 \dots n \end{aligned}$$

(9.7)

También igual que en el caso de clasificación se puede admitir que algunos puntos puedan caer fuera del espacio a  $\pm \varepsilon$ , modificando la restricción y agregando un costo que penalice los puntos que quedan fuera:

$$\min_{\beta, \beta_0} \frac{1}{2} \|\tilde{\beta}\|^2 + \gamma \sum_{i=1}^n (\zeta_i + \zeta_i^*) \quad (9.8)$$

*sujeito a :*

$$\begin{aligned} Y_i - \langle X_i, \tilde{\beta} \rangle + \beta_0 &\leq \varepsilon + \zeta_i, \quad \forall i = 1 \dots n \\ \langle X_i, \tilde{\beta} \rangle + \beta_0 - Y_i &\leq \varepsilon + \zeta_i^*, \quad \forall i = 1 \dots n \\ \zeta_i, \zeta_i^* &\geq 0 \end{aligned}$$

De igual forma que en el caso de clasificación este problema se puede resolver utilizando el Lagrangiano, KKT y el problema dual y se llega a la siguiente solución similar al caso anterior:

$$f(X) = \sum_{i=1}^n (\alpha_i^0 - \alpha_i^{0*}) \langle X_i, X \rangle + \beta_0^*$$

donde  $\alpha_i^0$ ,  $\alpha_i^{0*}$  son los multiplicadores de Lagrange correspondientes a las dos restricciones del problema.

Como se puede observar también, la solución sólo depende de los vectores del espacio a través de su producto interno por lo que se puede utilizar igual que en el caso de clasificación Kernels para resolver problemas de regresión no lineal y la función de regresión quedará:

$$f(X) = \sum_{i=1}^n (\alpha_i^0 - \alpha_i^{0*}) K(X_i, X) + \beta_0^*$$

donde igual que antes  $\beta_0^*$  se calcula a través de una de las condiciones KKT (la que se obtiene de los puntos que satisfacen la condición de estar sobre los hiperplanos), que también sólo depende del producto interno.

### 9.3. Estimador de la cola del enlace y del estado de un camino en la red

En el capítulo anterior se propuso utilizar como variable  $X$ , la distribución empírica del tiempo entre paquetes de prueba. En esta sección se propondrá un estimador  $X$  que presenta varias ventajas sobre el estimador anterior.

#### 9.3.1. El caso de un enlace

Si un paquete  $n$  arriba a la cola de un enlace en el instante  $t_n^i$  y sale de dicho enlace en  $t_n^o$ , siendo  $C$  la capacidad del enlace, entonces se cumple que:

$$q(t_n^i) + K = C(t_n^o - t_n^i)$$

Siendo  $K = P + DC$  constante, donde  $P$  es el tamaño fijo de los paquetes de prueba y  $D$  la latencia fija del enlace. Se podría usar esta ecuación directamente para estimar la cola del enlace, pero tiene una debilidad. El problema es que depende del tiempo en dos lugares diferentes y por lo tanto el sincronismo de ambos relojes afecta la medida. Por eso se buscará una ecuación que dependa sólo de la diferencias de tiempos en lugares con un mismo reloj.

Para comenzar se verificará que se cumple la siguiente ecuación:

$$q(t_n^i) = q(t_{n-1}^i) + C(t_n^o - t_{n-1}^o) - C(t_n^i - t_{n-1}^i)$$

Se sabe que  $q(t_{n-1}^i) + K = C(t_{n-1}^o - t_{n-1}^i)$ , por lo que sustituyendo en la ecuación anterior se llega a que:

$$q(t_n^i) = C(t_{n-1}^o - t_{n-1}^i) - K + C((t_n^o - t_{n-1}^o) - (t_n^i - t_{n-1}^i))$$

que reagrupando queda

$$q(t_n^i) + K = C(t_n^o - t_n^i)$$

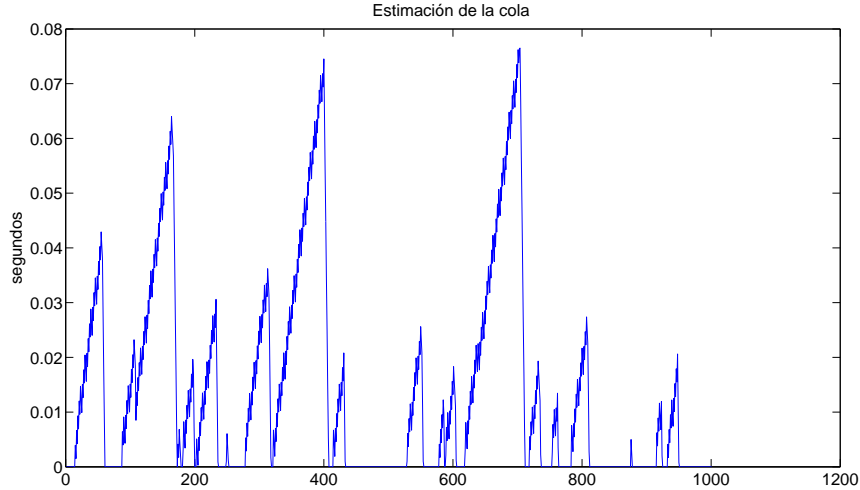


Figura 9.1: Estimación de la cola vista por cada paquete de prueba que arriba a la cola.

lo que se sabe es valor correcto de la cola en ese instante.

Aplicando la ecuación anterior de forma recurrente y asumiendo que hubo algún instante  $t_0^i$  donde la cola estuvo vacía (lo cual se sabe cierto si la cola es estable), entonces se llega a que:

$$\begin{aligned}
 q(t_n^i) &= \sum_{j=1}^{j=n} [C(t_j^o - t_{j-1}^o) - C(t_j^i - t_{j-1}^i)] \\
 &= C \sum_{j=1}^{j=n} [(t_j^o - t_{j-1}^o) - (t_j^i - t_{j-1}^i)]
 \end{aligned} \tag{9.9}$$

Esta ecuación permite estimar el valor de la cola del enlace en los instantes de arribo de los paquetes de prueba en función solamente de las diferencias de tiempo en puntos con el mismo reloj.

Utilizando la ecuación anterior se muestra en las figuras 9.1 y 9.2, la estimación de la cola para dos medidas con paquetes de pruebas de las simulaciones explicadas anteriormente.

Algunas observaciones:

1. Es necesario saber que la cola está vacía y de esa forma tener el punto  $t_0^i$  para hacer las medida. Para esto se debe observar que si la cola está vacía la diferencia de tiempo entre los paquetes a la salida del enlace es la misma que al ingreso de este. Esta sería una forma de detectarlo pero también podría pasar que con la cola en un período de ocupación, la cantidad de tráfico cruzado entre dos paquetes haga que el tiempo de salida coincida con el de entrada. Para solucionar esto en la práctica, se podría decidir que la cola está vacía si dos o más paquetes salen del enlace con la diferencia de tiempo igual a la diferencia de tiempo a la entrada. Esto se basa en la variabilidad del tráfico cruzado, ya que parece poco probable que durante varias



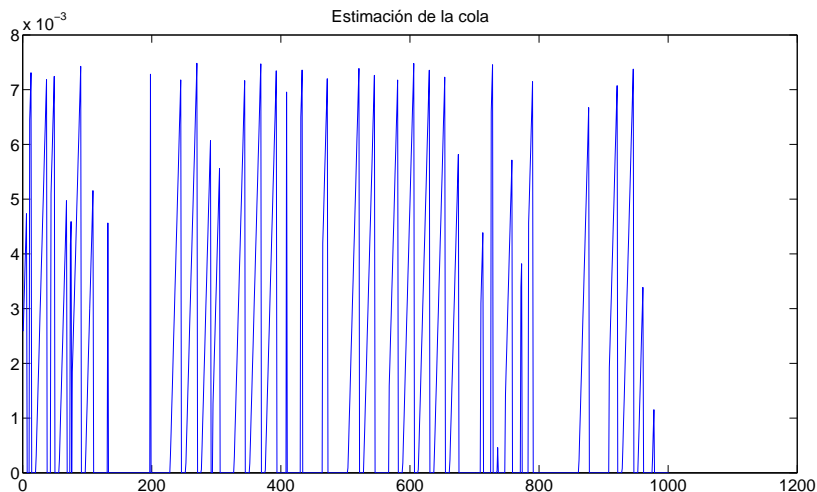


Figura 9.2: Estimación de la cola vista por cada paquete de prueba que arriba a la cola.

muestras la cantidad de tráfico cruzado acumulado entre los paquetes de prueba sea la misma, o al menos es más probable que si los tiempos entre varios paquetes a la entrada y a la salida del camino coinciden, esto se deba a la cola vacía y no al tráfico cruzado.

2. Lo anterior permite encontrar puntos donde la cola está vacía. Un algoritmo para estimar la cola puede comenzar por detectar estos puntos y luego a partir de estas zonas con cola vacía, reconstruir hacia adelante o hacia atrás el valor de la cola en los períodos de ocupación inmediatamente anterior o posterior. Esto además hace más robusto al algoritmo, ya que si hay errores de medida sistemáticos y estos se acumulan, pueden causar que el valor estimado de la cola no vuelva a cero en los períodos de cola vacía. De esta forma se evita que estos errores se trasladen de un período de ocupación al siguiente.
3. La ecuación anterior no sólo es válida cuando los tiempos entre partidas de paquetes de prueba son constantes sino que también lo es cuando estos siguen una distribución cualquiera.
4. En la ecuación anterior aparece la capacidad del enlace  $C$ . Esta capacidad puede ser conocida o no dependiendo de que el sistema de monitoreo de extremo a extremo sea administrado por el operador de la red o por otro agente externo. Si no fuera conocida, hay dos opciones: obtener el valor de la capacidad estimándola por alguno de los procedimientos conocidos, o simplemente si se asume constante y se va a utilizar en un método de aprendizaje estadístico como es el caso de este trabajo, utilizar solamente la diferencia de tiempos como estimador, que estima el tamaño de la cola dividido la capacidad del enlace.

### 9.3.2. Extensión a un camino en la red

En esta sección se supone que se tiene un camino en una red de  $N$  enlaces, cada enlace tiene capacidad  $C_l$  y se denominará  $q_l$  a la cola en dicho enlace.

Para cada enlace  $l \in (1 \dots N)$  del camino, la ecuación 9.9 dice que:

$$q_l(t_n^{i,l}) = C_l \left( \sum_{j=1}^{j=n} (t_j^{o,l} - t_{j-1}^{o,l}) - (t_j^{i,l} - t_{j-1}^{i,l}) \right) \quad (9.10)$$

sumando para todos los enlaces  $l = 1 \dots N$ , observando que  $t_j^{o,l} = t_j^{i,l+1}$ , y utilizando la siguiente notación:  $\Delta_{l,j} = t_j^{i,l} - t_{j-1}^{i,l} = t_j^{o,l-1} - t_{j-1}^{o,l-1}$ , se llega a que:

$$\begin{aligned} \sum_{l=1}^{l=N} q_l(t_n^{i,l}) &= C_1 \sum_{j=1}^{j=n} \Delta_{2,j} - C_1 \sum_{j=1}^{j=n} \Delta_{1,j} \\ &\quad + \dots + \\ &\quad C_l \sum_{j=1}^{j=n} \Delta_{l+1,j} - C_l \sum_{j=1}^{j=n} \Delta_{l,j} \\ &\quad + \dots + \\ &\quad C_N \sum_{j=1}^{j=n} \Delta_{N+1,j} - C_N \sum_{j=1}^{j=n} \Delta_{N,j} \end{aligned}$$

A continuación, se suma a cada fila  $C_N \sum_{j=1}^{j=n} \Delta_{l,j} - C_N \sum_{j=1}^{j=n} \Delta_{l+1,j}$  y a la última fila  $C_N \sum_{j=1}^{j=n} \Delta_{N,j} - C_N \sum_{j=1}^{j=n} \Delta_{1,j}$ . Observar que la suma total de estos términos que se agregan vale cero porque el segundo término de cada fila se cancela con el primero de la siguiente y el segundo de la última fila se cancela con el correspondiente de la primera. De donde se obtiene:

$$\begin{aligned} \sum_{l=1}^{l=N} q_l(t_n^{i,l}) &= (C_1 - C_N) \sum_{j=1}^{j=n} \Delta_{2,j} - (C_1 - C_N) \sum_{j=1}^{j=n} \Delta_{1,j} \\ &\quad + \dots + \\ &\quad (C_l - C_N) \sum_{j=1}^{j=n} \Delta_{l+1,j} - (C_l - C_N) \sum_{j=1}^{j=n} \Delta_{l,j} \\ &\quad + \dots + \\ &\quad C_N \sum_{j=1}^{j=n} \Delta_{N+1,j} - C_N \sum_{j=1}^{j=n} \Delta_{1,j} \end{aligned} \quad (9.11)$$

Por último teniendo en cuenta que  $\sum_{j=1}^{j=n} \Delta_{l+1,j} - \sum_{j=1}^{j=n} \Delta_{l,j} = \frac{q_l(t_n^{i,l})}{C_l}$  se llega a:

$$\sum_{l=1}^{l=N} q_l(t_n^{i,l}) \left(1 - \frac{C_l - C_N}{C_l}\right) = C_N \sum_{j=1}^{j=n} \Delta_{N+1,j} - C_N \sum_{j=1}^{j=n} \Delta_{1,j}$$

de donde,

$$\sum_{l=1}^{l=N} \frac{q_l(t_n^{i,l})}{C_l} = \sum_{j=1}^{j=n} \Delta_{N+1,j} - \sum_{j=1}^{j=n} \Delta_{1,j} \quad (9.12)$$

La ecuación anterior significa que sumando los tiempos entre llegadas de los paquetes de prueba hasta un cierto paquete y restando la suma de los tiempos entre partidas de los paquetes de prueba hasta dicho paquete, se obtiene una combinación lineal del estado de las colas que observó dicho paquete al pasar por cada una de ellas. Si todas las capacidades son iguales, lo que observa dicho paquete es la suma de la ocupación de todas las colas en el camino. Si no es así, obtiene una combinación lineal donde los coeficientes son el inverso de la capacidad de cada enlace.

La ecuación 9.12 muestra también que la media del tiempo entre paquetes a la entrada y a la salida del camino coincide ya que dividiendo entre  $n$  (número de paquetes de prueba) y haciendo tender  $n$  a infinito:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{l=1}^{l=N} \frac{q_l(t_n^{i,l})}{C_l} = \lim_{n \rightarrow \infty} \frac{1}{n} \left( \sum_{j=1}^{j=n} \Delta_{N+1,j} - \sum_{j=1}^{j=n} \Delta_{1,j} \right) \quad (9.13)$$

$$0 = \overline{\Delta_N} - \overline{\Delta_1} \quad (9.14)$$

donde  $\overline{\Delta}$  es el valor medio de  $\Delta$  y el primer término de la ecuación tiende a cero porque las colas son estables.

Esto justifica la imposibilidad de estimar  $Y$  utilizando como estimador  $X$  el promedio del tiempo entre paquetes.

## 9.4. Regresión utilizando el estimador de la cola o del estado del camino

El uso del estimador anterior que denominaremos  $\widehat{q}(t_n)$  ( $\widehat{q}(t_n) = \sum_{j=1}^{j=n} (t_j^{o,l} - t_{j-1}^{o,l}) - (t_j^{i,l} - t_{j-1}^{i,l})$ ) en lugar de utilizar el tiempo entre paquetes, tiene en cuenta al menos una de las debilidades mencionadas, que es la pérdida de información de correlación. Este estimador captura la información de la correlación del tráfico sobre la cola del enlace.

Se analizará en primer lugar, el uso del estimador de Nadaraya-Watson funcional usando como variable funcional  $X$  la distribución empírica de  $\widehat{q}(t_n^i)$ . Con esa estimación para el mismo conjunto de entrenamiento y validación de los ejemplos anteriores se obtiene el resultado de la figura 9.3.

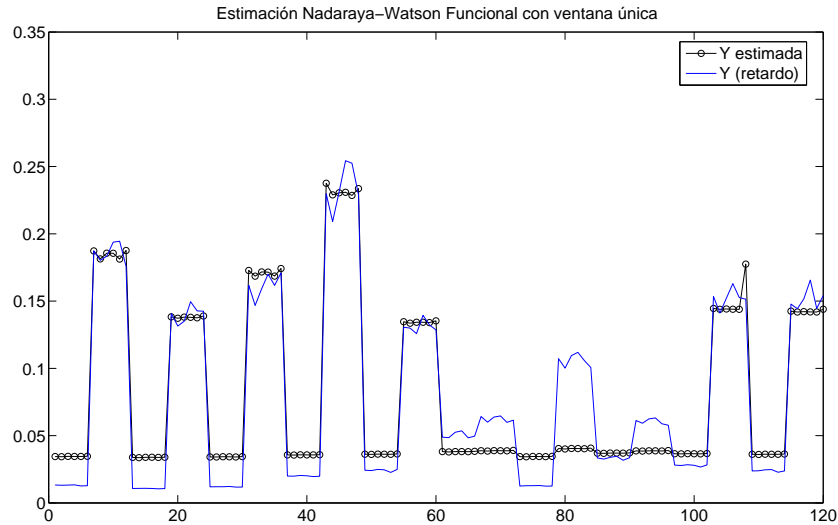


Figura 9.3: Nadaraya Watson funcional sobre la distribución de la cola.

Como se puede apreciar la aproximación es razonable en el caso de los picos y es muy mala para los valles. Esto conduce nuevamente al problema de la estimación de la ventana del núcleo. En este caso se estimó con una ventana única que minimiza el error cuadrático medio de toda la muestra de entrenamiento.

En las figuras 9.4 y 9.5 se muestra la estimación y el error relativo pero en este caso se utilizaron cuatro zonas para calcular la ventana como se explicó en la sección 8.4.2. En este caso la aproximación mejora sustancialmente y en casi todos los puntos se comete un error del 15 % a lo sumo.

A pesar de la buena aproximación lograda con 4 ventanas, llama la atención que se obtenga un resultado tan pobre en el caso de utilizar una única ventana, cuando se supone que esta estimación debiera dar resultados mejores.

La primera observación sobre este punto es la separación que existe entre los valores de las ventanas óptimas en la zona de menores valores de  $Y$  y en la zona de mayores valores de  $Y$ . En la zona de menores valores la ventana óptima encontrada es  $8 \times 10^{-5}$  y en la zona de valores mayores es de  $2 \times 10^{-3}$ , es decir que hay dos órdenes de magnitud entre ambos. Lo que sucede en el caso de usar una única ventana es que el valor óptimo era de  $1,4 \times 10^{-3}$  haciendo que con una ventana tan grande los puntos de los valles se estiman como un promedio de toda la zona baja.

Esta diferencia tan marcada en el valor de la ventana no se apreciaba en el caso utilizar la distribución del tiempo entre paquetes. Eso lleva a analizar cómo era la distribución del tiempo entre paquetes y la distribución de la cola.

En la figura 9.6 se muestran las distribuciones empíricas del tiempo entre paquetes para 4 muestras, dos de ellas corresponden a picos de  $Y$  y los otros dos a valles.

Como se puede observar en dicha figura las funciones distribución aún de muestras que presentan

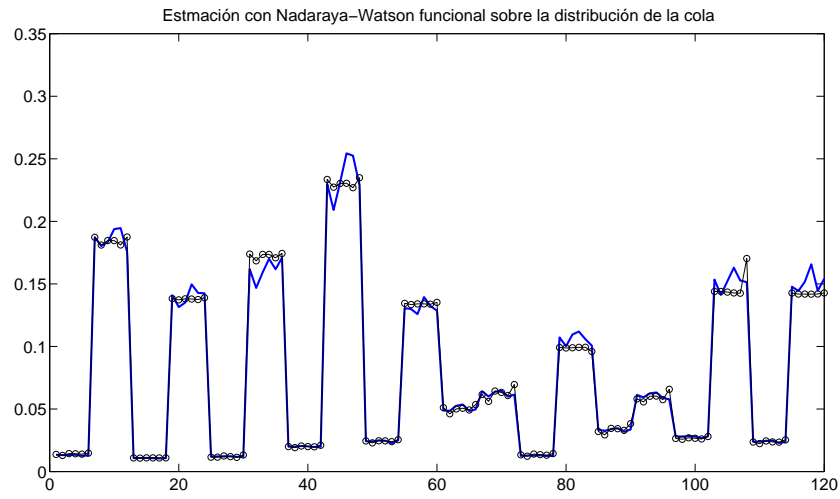


Figura 9.4: Nadaraya-Watson funcional con 4 zonas para la ventana .

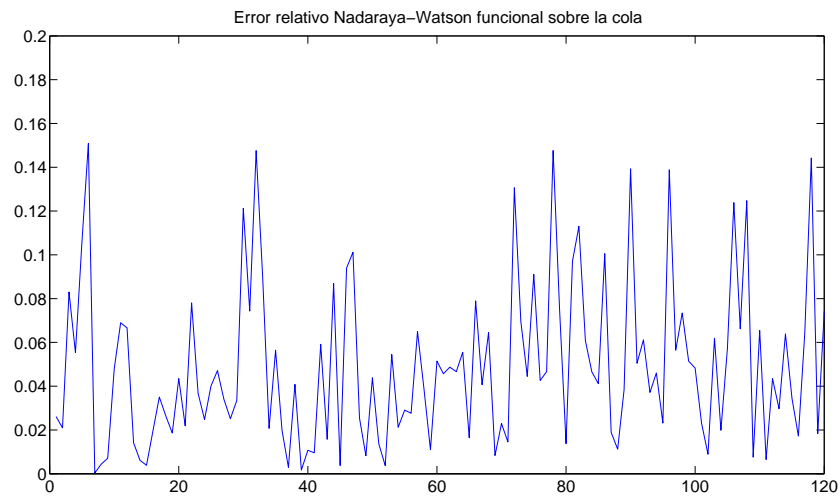


Figura 9.5: Nadaraya-Watson funcional con 4 zonas para la ventana .

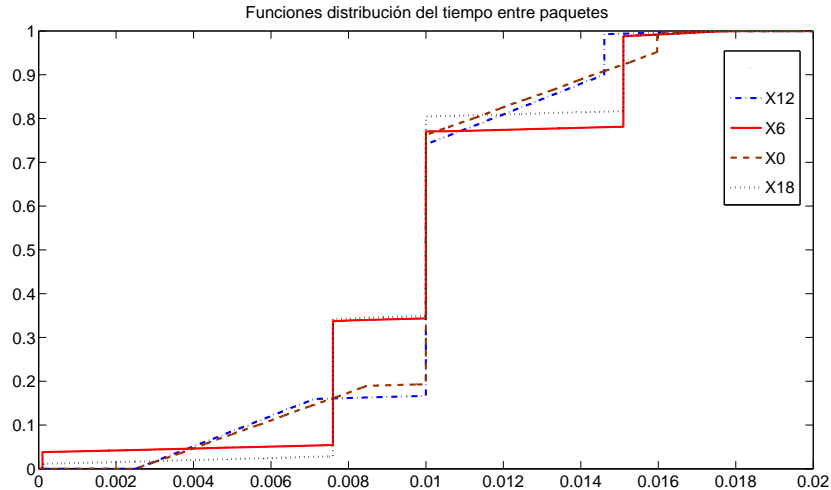


Figura 9.6: Distribución empírica del tiempo entre paquetes

valores de  $Y$  sustancialmente diferentes se diferencian poco. La norma  $L^1$  utilizada para medir la distancia entre estas distribuciones empíricas parece ser adecuada ya que entre ellas se distinguen por el área en ciertas zonas específicas.

Se analiza ahora la distribución empírica de la estimación de la cola en el enlace en la figura 9.7. En esta figura se pueden ver varias de estas distribuciones. En esta figura las 3 distribuciones de la derecha de la figura corresponden a valores de picos de  $Y$  y las restantes a valores medios y valles de  $Y$ . En la figura 9.8 se muestran sólo aquellas correspondientes a algunos valles y valores medios de  $Y$ . En este caso se puede apreciar que estas distribuciones se diferencian más fácilmente entre ellas. Sin embargo si se estima con la norma  $L^1$  esta distancia por esta misma separación se tienen valores muy grandes y valores muy pequeños de distancia. Esto explica porqué es necesario para poder tener una estimación razonable con Nadaraya-Watson hay que utilizar varios valores de ventana para diferentes zonas y sino los resultados son muy malos.

Una pregunta que surge de este análisis es si no se puede encontrar una distancia mejor que  $L^1$  para la distribución empírica de la cola. Este análisis de otras posibles distancias quedará como trabajo futuro. Una segunda pregunta es si con pocos parámetros bien elegidos de la distribución de la cola no será posible realizar una buena estimación, sin necesidad de comparar las distribuciones empíricas. Este último punto se analizará en la próxima sección.

## 9.5. Regresión con SVM utilizando parámetros de la estimación de la cola o del camino

Se propone en esta sección utilizar Support Vector Machines como método de regresión. Se utilizará este método por dos razones. En primer término, si se consigue hacer una regresión con pocos parámetros desde el punto de vista de la eficiencia tanto de almacenamiento como computacional

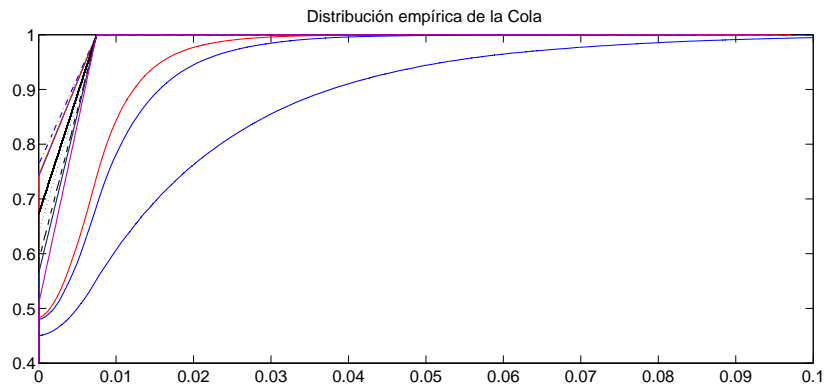


Figura 9.7: Distribución empírica de la cola

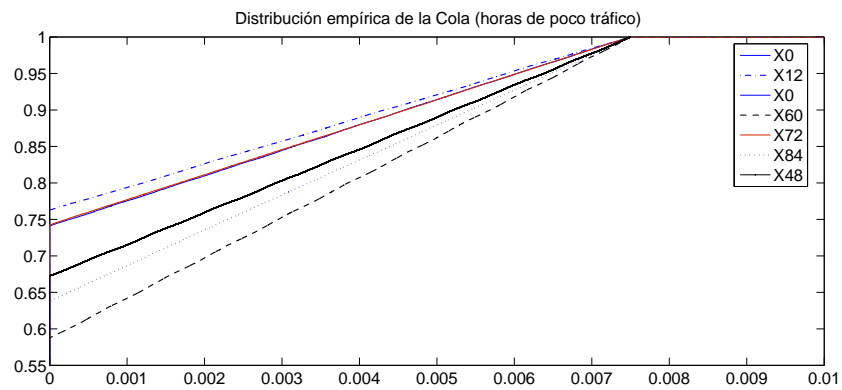


Figura 9.8: Distribución empírica de la cola sólo para los valles

es una ventaja importante para una aplicación en línea frente a tener que almacenar la función distribución y medir distancias sobre ésta.

En segundo lugar, se intentará utilizar SVM en lugar de Nadaraya-Watson por el siguiente motivo. Si con SVM se consigue tener buenas aproximaciones, este método presenta la siguiente ventaja. Nadaraya-Watson no requiere un modelo pero eso hace que la propia muestra de entrenamiento sea el modelo. Es decir que para cada nuevo punto a estimar hay que calcular la distancia con todos los puntos de la muestra de entrenamiento y hacer la estimación utilizando todos los puntos. En el caso de SVM, sólo se requieren los “support vector” y no toda la muestra de entrenamiento. Para muestras de entrenamiento grandes representa una ventaja tanto en almacenamiento de información como en costo computacional.

La primera pregunta es si se podrá tener una buena aproximación sólo con la media o con la media y la varianza de la estimación de la cola.

Se utilizará el mismo conjunto de muestras que en el caso de Nadaraya-Watson. Para las 360 observaciones  $(X, Y)$  de tráfico simulado, se separa la muestra en 2,  $(X^1, Y^1)$  con los valores de las observaciones para  $i = 1, \dots, 120$  (muestra de validación) y  $(X^2, Y^2)$  con los valores de las observaciones para  $i = 121, \dots, 360$  (muestra de entrenamiento). Se calcula el estimador  $\phi_n(X_i^1)$ , para cada  $i = 1, \dots, 120$ , a partir de las 240 observaciones de la segunda muestra.

En este caso no hay que ajustar la ventana de Nadaraya-Watson, pero hay que ajustar los parámetros del modelo de SVM. El parámetro  $\gamma$  (ecuación 9.8 por ejemplo, controla el compromiso entre la minimización del error y la maximización del margen. Otros parámetros dependen del núcleo que se utiliza para hacer el mapeo no lineal. En este caso se utilizará un núcleo de base radial 9.4, por lo que en este caso también hay que ajustar el parámetro  $C$ . El método más simple para hacer este ajuste es considerar una grilla sobre el espacio de parámetros y buscar el punto que minimiza el error cuadrático medio sobre la muestra de validación. Existen algoritmos más eficientes, pero en nuestro caso en los ejemplos manejados se vio que existía bastante robustez ante pequeños cambios en los parámetros y además que el tiempo de cómputo con una grilla daba resultados razonables.

En la figura 9.9 se puede ver como la estimación utilizando SVM tomando como variable  $X$  sólo la media, da una estimación pobre, la cual mejora cuando se agrega la varianza aunque todavía existen algunos puntos donde la estimación no es buena como en el señalado con un círculo en dicha figura.

En las distribuciones empíricas que se muestran en las figuras 9.7 y 9.8, se puede apreciar que porcentaje de tiempo de cola vacía parece discriminar bien entre las distribuciones, sobre todo para los valores de  $Y$  correspondientes a los valles. Por otro lado, también se puede ver que para las distribuciones correspondientes a valores altos de  $Y$  se discriminan bien por los percentiles cercanos a 1 de la distribución. Estas observaciones parecen tener sentido en general, para niveles bajos de tráfico la cola se llena poco y lo que diferencia es cuánto tiempo pasa vacía y cuánto tiempo tiene algún paquete. En cambio para valores de tráfico alto lo que pasa a determinar la diferencia es cuanto alcanza llenarse la cola.

Por este motivo se decidió agregar a la media y la varianza de la cola estos dos nuevos parámetros: el porcentaje de tiempo de que la cola está vacía y el percentil 90 %. En la figura 9.10 se muestra la estimación realizada con SVM utilizando sólo estos cuatro parámetros. En la figura 9.11 se muestra el error relativo que se comete en la estimación. Como se puede ver la aproximación es buena



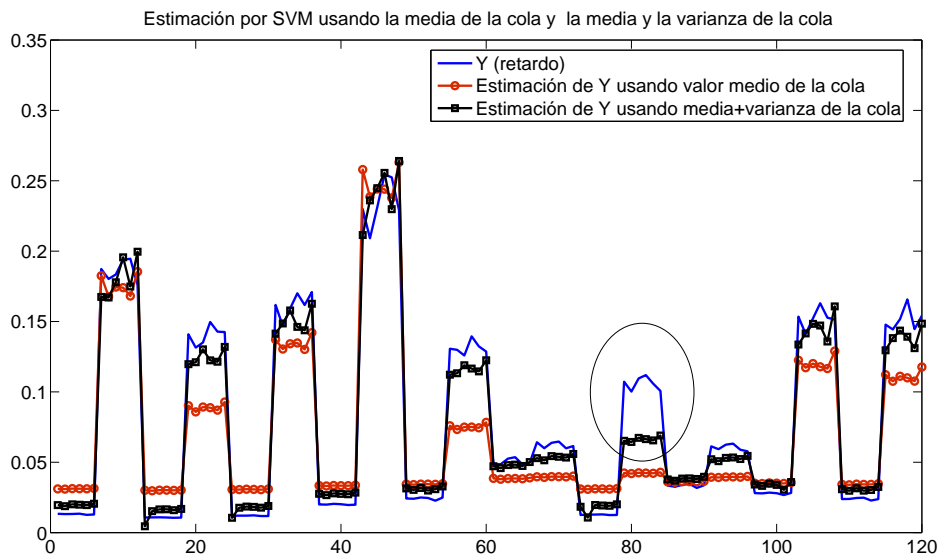


Figura 9.9: Estimación con media y media/varianza de la cola

y comparable a los resultados obtenidos utilizando Nadaraya-Watson funcional utilizando como variable  $X$  la distribución empírica de la cola. Una observación adicional es que los puntos donde se da una peor aproximación (aproximadamente 20%) es en uno de los valles donde desde el punto de vista práctico a los efectos de estimar la QoS es menos importante el error.

## 9.6. Estimación con diferentes videos

En esta sección se analizará la estimación de los parámetros de desempeño de un video, pero habiendo entrenado con otro video diferente. El objetivo es verificar que no es necesario entrenar el sistema para cada video específico, sino que la estimación de un video es razonablemente buena si se entrena con otro video de características similares (por ejemplo con el mismo tipo de codec, tasa de frames y cantidad de movimiento). En el escenario de simulación utilizado en las secciones anteriores, se midió el retardo del video usando trazas de aproximadamente 30 segundos de diferentes videos. Específicamente lo que se hizo fue modificar las simulaciones con ns-2, y en lugar de generar tráfico de video sintético (utilizando un proceso ON-OFF), utilizar las trazas de diferentes videos. Se generaron para cada video diferente 200 muestras  $(X, Y)$  donde  $X$  es el estimador de la cola del enlace obtenido utilizando los paquetes de prueba e  $Y$  es el retardo medido sobre el tráfico de video. De las 200 muestras se tomaron 120 como muestras de entrenamiento y 80 para validación. Posteriormente utilizando SVM, se estimó el retardo de las 80 muestras de validación para cada uno de los videos utilizando dos muestras de entrenamiento diferentes. Por una lado se utilizó la muestra de entrenamiento del video correspondiente y por otro, la muestra de entrenamiento correspondiente a un video tomado como referencia.

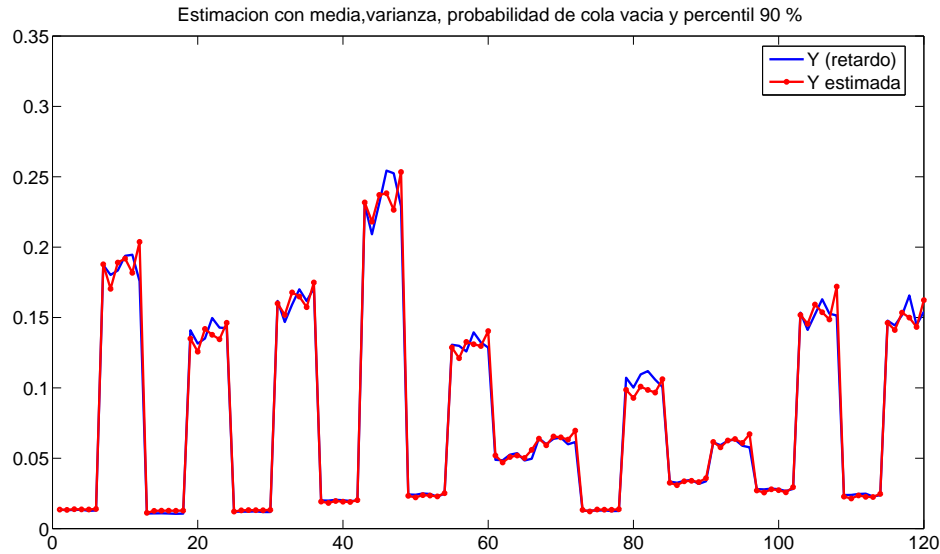


Figura 9.10: Estimación con 4 parámetros de la distribución del estimador de la cola

En el cuadro 9.1 se muestran las características de 4 videos utilizados. Como se puede ver los primeros 3 videos fueron codificados con el mismo codec y con la misma tasa de frames. Los tres videos son partes de filmaciones diferentes. El video 1 corresponde a la filmación de una descarga eléctrica en una central eléctrica, el video 2 corresponde a la filmación de un salto desde un puente, y el video 3 al desplazamiento de un auto en una carretera. El video 4 está codificado con otro codec y con otra tasa de frames y corresponde a la filmación de una persona hablando.

Cuadro 9.1: características de los videos utilizados

Video	Codec	tasa de frames
1	MPEG 1	25 fps
2	MPEG 1	25 fps
3	MPEG 1	25 fps
4	MPEG 4	30 fps

Como se puede ver en la figuras 9.12 y 9.13, la estimación del retardo de la muestra de validación del video 2 y del video 3, realizada con la muestra de entrenamiento del mismo video (video 2 y video 3 respectivamente) y la estimación de cada uno de ellos realizada con la muestra de entrenamiento del video 1 son todas razonablemente buenas. Incluso en el caso del video 2, la estimación realizada entrenando con el video 1 es en algunos casos (en los puntos con retardo en el entorno de 0.4 segundos) mejor que la realizada entrenando con el propio video 2. Esto sin embargo es fortuito y se debe a que en la muestra de entrenamiento hay pocos valores en ese rango de retardo y la



Figura 9.11: Error relativo

estimación por tanto no es del todo buena en ese rango. Por otro lado, en la figura 9.14 se puede ver que la estimación realizada entrenando con el propio video 4 es razonablemente buena, pero la realizada entrenando con el video 1 es mala. Esto era esperable ya que el video 4 tiene características en cuanto al codec utilizado y la tasa de frames diferentes a los del video 1.

## 9.7. Conclusiones

En este capítulo se evalúan alternativas para mejorar el método de estimación de la QoS de extremo a extremo visto en el capítulo anterior. Se propone utilizar un nuevo estimador del estado de la red, un estimador que brinda información sobre la cola de un enlace o las colas del camino. Ese estimador brinda más información sobre el estado de la red que el estimador de la distribución empírica de los tiempos entre paquetes utilizado en el capítulo anterior. Este aumento de información proviene de que el estimador de la cola recupera información de correlación que se pierde con el anterior. Además, este estimador permite estimar con un buen grado de aproximación la QoS de extremo a extremo de una aplicación utilizando sólo algunos parámetros de su distribución, sin necesidad por lo tanto de hacer una regresión funcional. Este resultado habilita además, a realizar una regresión utilizando Support Vector Machines en  $\mathbb{R}^d$ . SVM presenta la ventaja frente a Nadaraya-Watson que requiere almacenar menos información de entrenamiento y en consecuencia tiene un menor costo computacional para calcular cada nueva estimación. Esta última observación es muy importante si se desea realizar aplicaciones en línea.

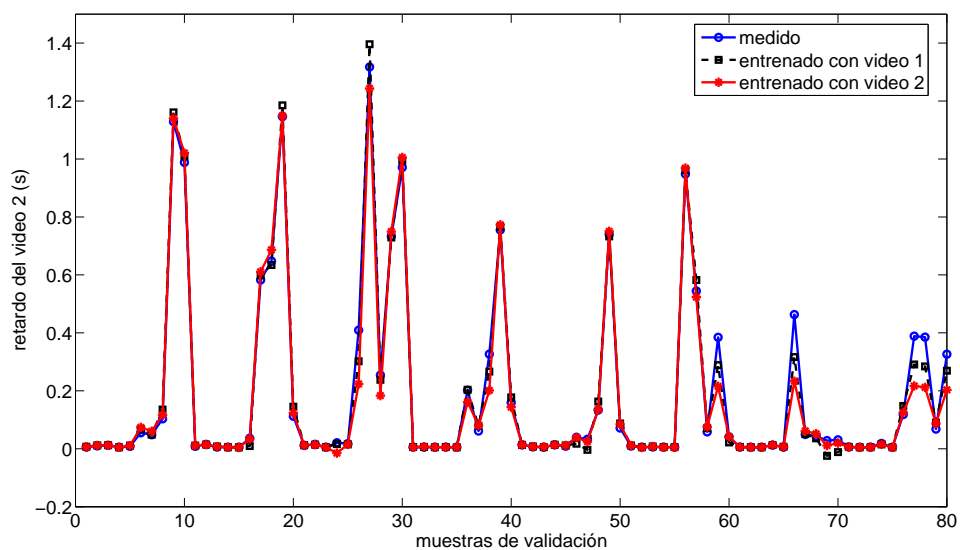


Figura 9.12: valor medido y estimado del retardo del video 2

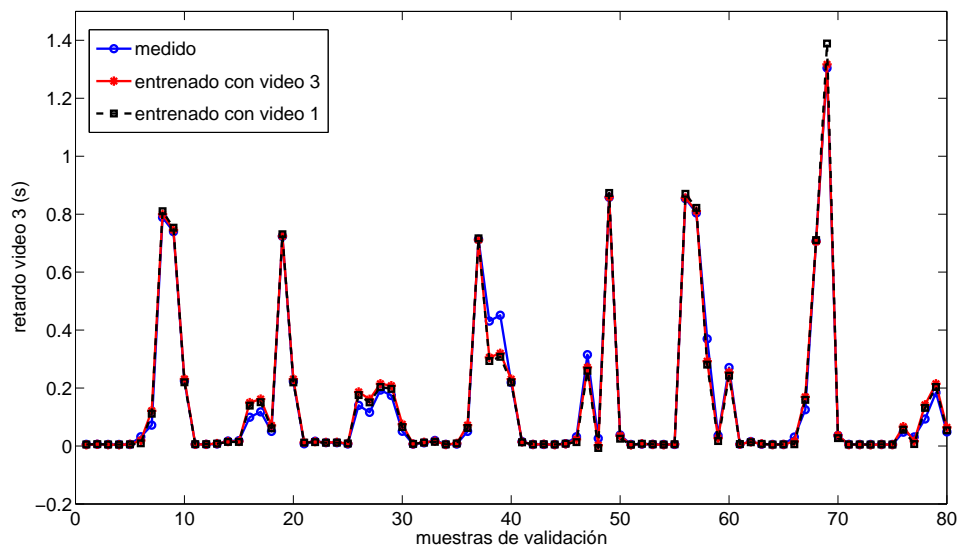


Figura 9.13: valor medido y estimado del retardo del video 3

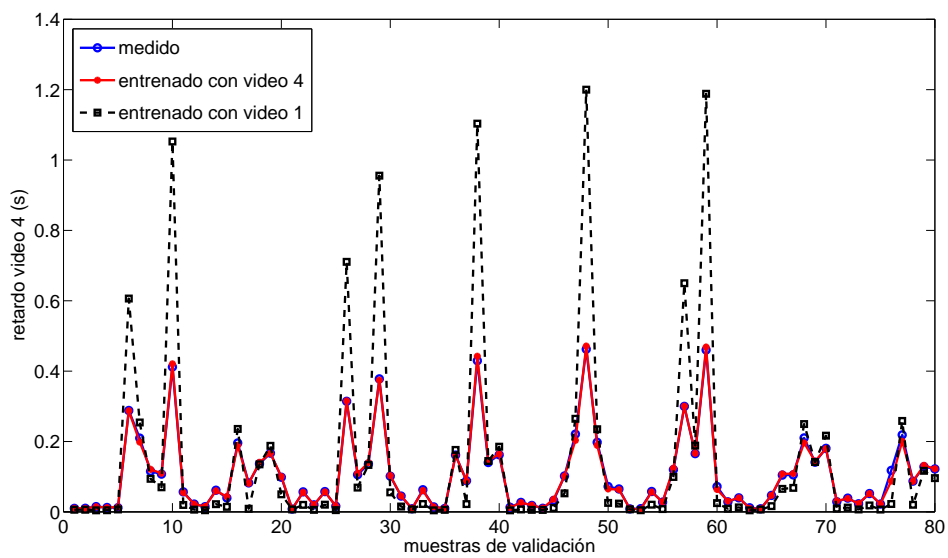


Figura 9.14: valor medido y estimado del retardo del video 4

## Capítulo 10

# Software de medición y estimación de la QoS

### 10.1. Introducción

Este capítulo describe el desarrollo de dos herramienta de software (MetroNet y MetroRed) que permiten realizar mediciones de la calidad de servicio de extremo a extremo en Internet. Estas herramientas, si bien están orientadas para la realización y análisis de mediciones de extremo a extremo en una red, implementan además las técnicas de estimación desarrolladas en la tesis. Con ellas se pueden enviar paquetes de prueba, videos de entrenamiento y tienen incorporados los métodos de regresión vistos para estimar la calidad de servicio.

Las dos herramientas mencionadas fueron probadas primero en una red piloto de máquinas Linux y con tráfico simulado, en una situación controlada donde los resultados pueden ser fácilmente contrastados con las mediciones reales. Por último se han realizado pruebas en Internet para evaluar su operatividad y probar la eficiencia de los algoritmos desarrollados en una red real.

MetroRed es una evolución de MetroNet que introduce cambios importantes en la arquitectura del software y en el diseño del mismo. En MetroNet participaron también en el desarrollo: Pedro Casas, Víctor González y Federico Larroca. Ellos tuvieron un aporte fundamental en el desarrollo de la interfaz de usuario, en el diseño y programación de la base de datos y en la integración de los diferente módulos. En el caso de MetroRed el desarrollo es propio. En este caso la interfaz de usuario es desde línea de comandos ya que por razones de diseño se eliminó el modelo WEB cliente servidor de MetroNet como se verá más adelante y aún no se desarrolló una interfaz gráfica.

La arquitectura de MetroNet y algunos resultados de su aplicación a redes reales fueron publicados en [19]. En el caso de MetroRed al momento que se escribe esta tesis aún no han sido enviados para su publicación.

### 10.2. Metronet

Metronet tuvo como objetivo encontrar una metodología y una herramienta informática para realizar medidas de extremo a extremo de los diferentes parámetros de calidad de servicio para

diferentes aplicaciones. El software si bien plantea un marco general de mediciones de extremo a extremo, permite implementar las metodologías de estimación de QoS vistas en los capítulos anteriores. Esta sección describe el producto de software obtenido, sus prestaciones, gestión de desarrollo y arquitectura.

### 10.2.1. Especificación del software: restricciones y requerimientos.

En el momento de la especificación del software existían algunas restricciones, formuladas en los objetivos del proyecto, orientadas esencialmente en dos sentidos:

- Facilitar al usuario el acceso al software y su operación;
- Cargar lo menos posible la máquina del usuario, atendiendo a la diversidad de hardware, software instalado y estado de procesos. Esta restricción tiende a la universalidad en el uso del software y a una mínima incidencia del estado de la máquina cliente sobre la medición.

Más específicamente, el software debía cumplir las siguientes restricciones (o requerimientos no funcionales):

- Escrito en un lenguaje de programación libremente accesible para el usuario.
- Carga mínima en la máquina del usuario, limitada a la devolución de los paquetes enviados por el ISP con mínimo trámite.
- Descarga transparente del código cliente necesario, sin intervención específica del usuario.
- Herramientas de la máquina del usuario limitadas a un navegador y eventualmente una máquina virtual para la ejecución del código cliente.

Los requerimientos funcionales del software eran los siguientes:

- Facilidad de uso.
- Interfaz clara, sobria, sin excesivo detalle técnico, tanto en la especificación de las pruebas como en la presentación de resultados.
- Validación de acceso del usuario, mediante nombre y contraseña.
- Posibilidad del usuario de fijar localizaciones y ser reconocidas éstas luego, para poder efectuar mediciones desde diferentes lugares y utilizar datos de verificaciones anteriores como insumo para la evaluación estadística.
- Posibilidad del usuario de elegir tipo de medio (audio o video).
- Presentar valores por defecto para todos los parámetros, siempre que ello sea posible, permitiendo al usuario fijar los suyos propios si lo desea.

- Posibilidad del usuario de elegir características del multimedia, tales como tipo de codec, tasa de bits, tipo de conexión, cantidad de movimiento (video). Se presentan sólo aquellos valores correspondientes al tipo de multimedia elegido para la verificación.
- Posibilidad del usuario de elegir características del experimento, fijando cantidad de pruebas, tiempo entre pruebas, cantidad de paquetes de prueba, tiempo entre paquetes.
- Opcionalmente, el usuario puede pedir la transmisión de una muestra del medio elegido para una evaluación subjetiva, de entre una lista ofrecida.

Si bien no forma parte de la metodología de medida, se estimó de gran interés para el usuario ofrecer la posibilidad de elegir una muestra de video y verla por sí mismo, habilitando para que en un futuro se pueda integrar la estimación subjetiva de la calidad. La lista de videos ofrecida intenta abarcar los tipos más frecuentes en el medio elegido, posibilitando al usuario elegir aquel que más se aproxima al medio que realmente quiere ver y oír. Las muestras tienen una duración típica de 20 a 30 segundos.

Se decidió realizar la implementación del software en Java [138], por su independencia de plataforma y su ubicuidad. El código cliente se implementó como “applet” [139], y sólo devuelve los paquetes recibidos de la dirección del ISP a esa misma dirección, con un mínimo tratamiento de estampado de fecha y hora. El hecho de realizar estas mediciones en capa de aplicación se consideró pertinente, por simular más de cerca la situación real del usuario al recibir voz o video, lo cual se realiza siempre mediante una aplicación.

### 10.2.2. Uso del software.

El usuario dirige su navegador a la dirección URL del proveedor cuya calidad de transmisión quiere evaluar. La primera página recibida es una bienvenida al programa donde se le pide usuario y contraseña, o se le ofrece bajar el software necesario (applet de Java) para instalar en el cliente, si no lo ha hecho ya.

**MetroNet**

Software para estimación de Calidad de Servicio  
en transmisión multimedia sobre IP

Nombre de usuario

Contraseña

Proyecto PDT S/C/OP/17/02  
Instituto de Ingeniería Eléctrica  
Facultad de Ingeniería - UDELAR

Figura 10.1: Bienvenida y autenticación del usuario.



Una vez validado, el usuario recibe una página donde se fijan todos los parámetros del experimento. El usuario debe elegir obligatoriamente la localización, si se trata de audio o video; el resto de los parámetros ofrecen valores por defecto que el usuario puede simplemente aceptar. Si desea recibir una muestra real de audio o video, deberá elegirla entre la lista de muestras disponibles.

Figura 10.2: Ingreso de parámetros del experimento

En el contexto de este software el término “experimento” se aplica al proceso de medición completo, el cual consta de varias “pruebas”. Esto resulta aparente en el propio contenido y distribución de la página, por lo que el manejo debiera ser totalmente intuitivo.

En cuanto el usuario decide comenzar la prueba, en el servidor del ISP se buscan registros históricos para ese usuario y localización, si existen, se evalúa su utilidad, diseñando el patrón de pruebas en consecuencia. El servidor emite paquetes de sondeo o paquetes de video, el cliente los devuelve, se miden tiempos, se contabilizan pérdidas, elaborando todos estos datos y guardándolos en una base de datos.

Si el usuario pidió una muestra del medio, se le envía para su audición y visualización en el propio navegador. Se presenta finalmente una página de resultados con la evaluación realizada. El usuario puede repetir el experimento, en cuyo caso vuelve a la página de parámetros, donde encuentra los últimos fijados, o terminar la aplicación. Pasado un cierto tiempo, el servidor cierra la sesión.

### 10.2.3. Arquitectura y diseño de MetroNet.

La arquitectura es un modelo cliente/servidor clásico: quien quiere medir la calidad de servicio actúa como cliente, sólo devolviendo paquetes; el proveedor de servicio actúa como servidor, autenticando al usuario, emitiendo los paquetes, recibiendo las devoluciones, realizando los cálculos y devolviendo al cliente los resultados.

La figura 10.3 muestra el despliegue de paquetes en el cliente y el servidor. El nodo cliente es realmente liviano; la carga de procesamiento, memoria y repositorio de datos aparece soportada en el nodo servidor.

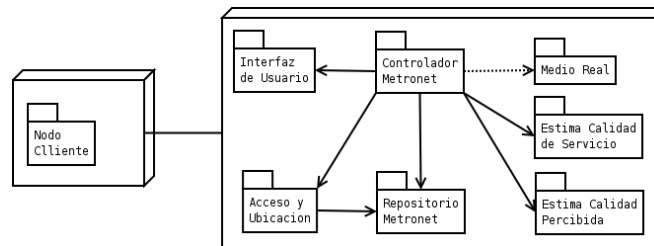


Figura 10.3: Diagrama de Componentes en cliente y servidor.

Seguidamente se revisan brevemente algunos aspectos particulares de la concepción del software.

La figura 10.4 muestra los paquetes en el nodo cliente. El paquete Rebotador es el applet de Java cuya función esencial es devolver al proveedor los paquetes que éste le envía, ya sean de sondeo o de video simulado, visualizadas en el detalle de análisis en la misma figura.

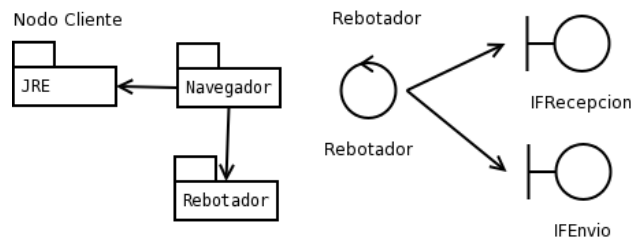


Figura 10.4: Nodo cliente: diagrama de paquetes y detalle de análisis del rebotador.

El diagrama de análisis de la figura 10.5 muestra el software de Metronet en un único diagrama, destacando la arquitectura de tres capas, con las interfaces sobre la izquierda, el repositorio (DBMS) en la parte inferior, el control principal del software, Estimador Flujo MMedia, en el centro, y el resto de las funciones como paquetes controlados por Estimador Flujo MMedia. Esta separación resultó particularmente útil en el caso de los dos paquetes de estimación (Estima Calidad de Servicio y Estima Calidad Percibida), donde son más probables los cambios.

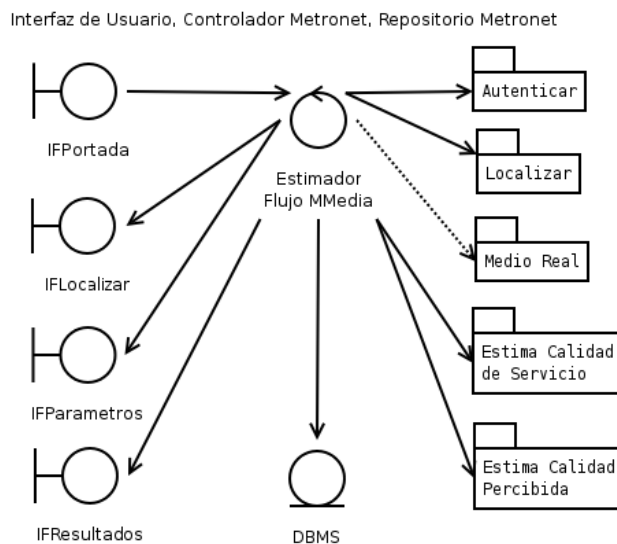


Figura 10.5: Arquitectura de tres capas.

El diagrama de la figura 10.6 muestra el desacoplamiento entre los métodos de medida y las herramientas de medición y cálculo. Es relativamente fácil agregar nuevas herramientas, así como nuevos métodos en base a herramientas existentes.

El método de medida Medio con Histórico se aplica cuando el usuario ya ha realizado experimentos desde esa misma ubicación, existen datos de medidas almacenados, y se estima que esos datos son relevantes para la situación actual de la red, en cuyo caso se utilizan, reduciendo las pruebas de envío y recepción de paquetes. Si los datos históricos no existieran, no fueran suficientes o resultaran inadecuados para la situación actual de la red, se deriva automáticamente al método de medida Medio sin Histórico, basado netamente en pruebas de envío y recepción de paquetes para el momento actual.

La figura 10.7 muestra las clases principales con las que se implementaron las herramientas de medida, así como algunos de los métodos.

### 10.3. MetroRed, una red overlay de mediciones distribuida

MetroRed es una evolución de MetroNet pero que cambia diversos aspectos fundamentales de la arquitectura. El primer punto sustancial es eliminar la arquitectura cliente/servidor. MetroRed pasa a tener una arquitectura de nodos distribuidos que forman una red overlay de mediciones. Estos nodos se despliegan en diferentes equipos en los que interesa registrar medidas. Los nodos son todos idénticos pero pueden adoptar diferentes comportamientos de acuerdo a la configuración de cada experimento. Los experimentos se configuran en un coordinador. En esta configuración

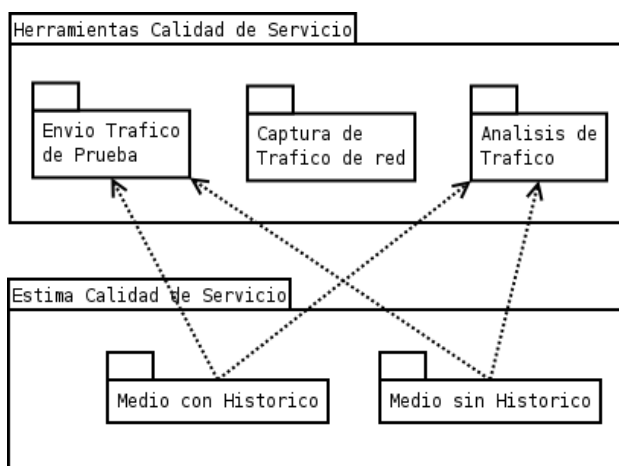


Figura 10.6: Métodos y herramientas: separación y relaciones.

se especifica una agenda de los diferentes experimentos a realizar y en qué momentos deben ser realizados.

Además se indica qué nodos participarán y cuál es la topología de la red overlay de mediciones en cada experimento (cómo se “enlazan” los nodos que participan en el camino overlay utilizado en ese experimento). Esto modificó sustancialmente la arquitectura del sistema de mediciones. Además al no existir la arquitectura cliente/servidor ya no existe una aplicación WEB servidora y un applet. Ahora los nodos son aplicaciones que un usuario corre en su máquina y tiene la misma jerarquía que cualquier otro nodo.

Otro aspecto importante es que se desacopló aún más la realización de los experimentos y el procesamiento de los datos colectados en un experimento. Es decir existen en esta nueva arquitectura dos módulos totalmente separados, el módulo de mediciones que es una herramienta general para hacer mediciones activas y monitorear de manera distribuida una red y el módulo de estimación de la QoS que tiene diferentes algoritmos para procesar las mediciones.

Se agregaron además nuevos algoritmos, se incorporó por ejemplo la posibilidad de utilizar Support Vector Machines.

El esquema de funcionamiento del software MetroRed (ver fig. 10.8) está formada por una red de nodos overlay que pueden generar, reenviar o recibir tráfico. Estos nodos pueden generar diferentes patrones de tráfico. Existe además un coordinador donde se agendan las pruebas y medidas a realizar. En el coordinador puede configurarse por ejemplo que cualquier nodo de la red de nodos de MetroRed genere tráfico para realizar mediciones. También puede configurarse la topología overlay de cada experimento. Esto da flexibilidad para realizar medidas a través de diferentes caminos de la red overlay de medición. En cada nodo se registra el tiempo en que el paquete pasa por el nodo por lo tanto, se puede analizar el resultado de extremo a extremo de un camino overlay o los resultados de un camino particular. A cada paquete de medida, en todos los nodos por los que pasa, se le estampa el tiempo de pasaje. Además, esta información puede ser registrada en un archivo en cada nodo. También pueden agendarse pruebas que se realicen periódicamente. La topología overlay de medida puede cambiar en cada prueba y esto se configura también en el coordinador.

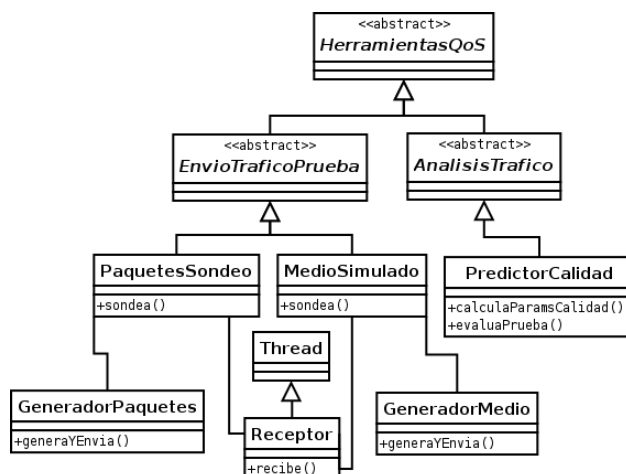


Figura 10.7: Herramientas para calidad de servicio, clases principales.

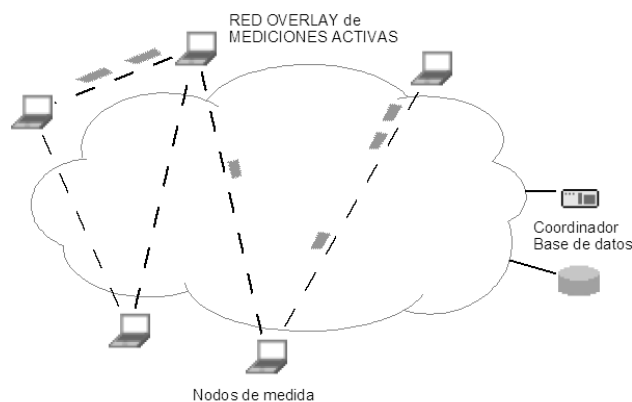


Figura 10.8: Esquema MetroRed

Como se mencionó, en el software están totalmente desacopladas dos funciones:

- El sistema de mediciones, que incluye: la generación de tráfico de diferentes tipos, la realización de pruebas a través de la red overlay de medidas y el registro de la información al atravesar la red de los paquetes generados.
- El procesamiento de esa información para estimar diferentes parámetros relacionados con la QoS, que incluye: la estimación de la capacidad del cuello de botella de un camino, la estimación de la QoS de una cierta aplicación, etc.

El sistema de mediciones registra en una base de datos o en archivos planos la información generada por las mediciones. El sistema de procesamiento toma esta información y mediante diferentes algoritmos la procesa.

Una de las funcionalidades implementadas en el módulo de estimación de la calidad de servicio de MetroRed es la desarrollada en la tesis, que permite estimar la QoS de una aplicación a partir de tráfico de prueba. Esto se hace a través de la implementación de algoritmos de aprendizaje estadístico ya sea utilizando Nadaraya-Watson (funcional o en  $\mathbb{R}^n$ ) o Support Vector Machines. Pero este es un algoritmo más, no estando ligado MetroRed a la utilización de esta metodología.

### 10.3.1. Paquetes que componen MetroRed

Como se mencionó antes, MetroRed está compuesto de dos módulos desacoplados. En la figura 10.9 se muestran estos módulos y los paquetes que componen cada uno. El módulo de Mediciones-MetroRed es responsable de efectuar los experimentos de medición. El módulo de EstimacionQoS es el responsable de procesar los experimentos realizados. A continuación se describen brevemente cada uno de los paquetes, haciendo énfasis en aquellas cosas que cambian significativamente respecto de MetroNet.

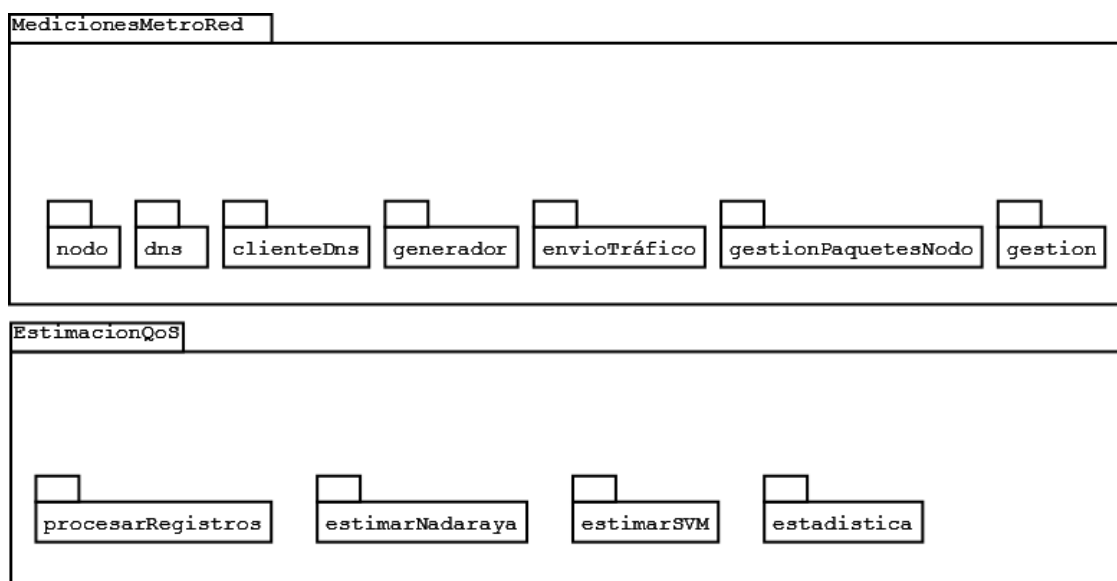


Figura 10.9: Módulos y paquetes que componen MetroRed.

#### MetroRed:dns

El paquete dns es el responsable de mantener un servicio de nombres de MetroRed. La idea es que cada nodo en MetroRed se identifica mediante un número o nombre (Id). Esta identificación es su dirección en la red overlay. En la configuración de un experimento, de su topología, etc. los nodos se identifican sólo por su Id. La idea detrás de esto es desacoplar su identificación en la red overlay, de su dirección en la capa de red (su ip), que en algunos casos puede variar con el tiempo,

en otros casos puede suceder que un nodo tenga más de una ip asociada, etc. El servicio de dns está corriendo siempre y los nodos se comunican con él para registrar la dirección de capa de red. Los nodos también pueden consultar el dns para saber en qué ip encuentran a un determinado nodo en cierto momento. Además de la dirección ip en el dns se registran los puertos que utiliza el nodo para recibir/enviar tráfico UDP/TCP. De esa forma al consultar el dns se puede saber en/desde qué ip y puerto un nodo en determinado momento escucha/envía cierto tipo de paquetes. El hecho de registrar los puertos además de la ip, tiene la función (además de dar flexibilidad al sistema), de que los nodos puedan correr detrás de un NAT (Network Address Translation). Al estar detrás de un NAT para enviarle tráfico es necesario conocer la ip pública del NAT y el puerto que el NAT asoció al nodo, además de que el nodo que envía tráfico debe estar registrado en el NAT. Para tal fin cuando el nodo se conecta al dns, este registra también la dirección ip pública del NAT y el puerto del NAT desde donde se envió el paquete. Esta información es la que luego se le comunica a un nodo que quiera enviarle tráfico al nodo en cuestión. El nodo al enviar el paquete de registro al dns lo debe hacer desde el mismo puerto desde el que luego escuchará para recibir tráfico de otros nodos. En la enorme mayoría de los NAT, cuando un nodo envía tráfico UDP hacia afuera del NAT, el NAT asocia un puerto que está relacionado con el puerto que utilizó el nodo. Cuando desde afuera se envía tráfico a la ip del NAT y a dicho puerto, el NAT lo reenvía al nodo y al puerto desde donde inició el nodo el tráfico. Sin embargo, si el tráfico viene de una ip diferente de la dirección a la que se envió tráfico desde el nodo, este tráfico entrante no será reenviado al nodo por el NAT. Para que sea reenviado el tráfico entrante, la ip del originador debe estar registrada en el NAT. Por este motivo cuando un nodo va a recibir tráfico de otro nodo, debe enviar algunos paquetes al nodo del que quiere recibir. El objetivo de estos paquetes es abrir el NAT para la ip del nodo que le va a enviar paquetes (en realidad si el otro nodo está detrás de otro NAT los paquetes los envía a la ip pública de este segundo NAT). Muchos servidores NAT operan de la manera descrita. Hay algunos NAT que no operan de esta forma. En esos casos no es posible utilizar MetroRed detrás de ellos. En la figura 10.10 se muestra el diagrama de clases de este paquete.

### **MetroRed:clienteDns**

Este paquete es el responsable de:

- Informar al dns de la ip y puertos que utilizará el nodo al cual este cliente está asociado.
- Consultar al Dns la ip y puerto en que escucha el nodo al que tiene que enviarle tráfico en el experimento el nodo asociado a este clienteDns.
- Consultar al Dns la ip y puerto desde los cuales otro nodo le enviará tráfico al nodo asociado a este clienteDns.
- Abrir el NAT para poder recibir paquetes del nodo que le enviará tráfico en el experimento. Para este fin, usando los datos del punto anterior, enviará paquetes de prueba a la ip y puerto del nodo que le enviará tráfico en el experimento para registrar esa ip en su NAT.

En la figura 10.11 se muestra el diagrama de clases de este paquete.

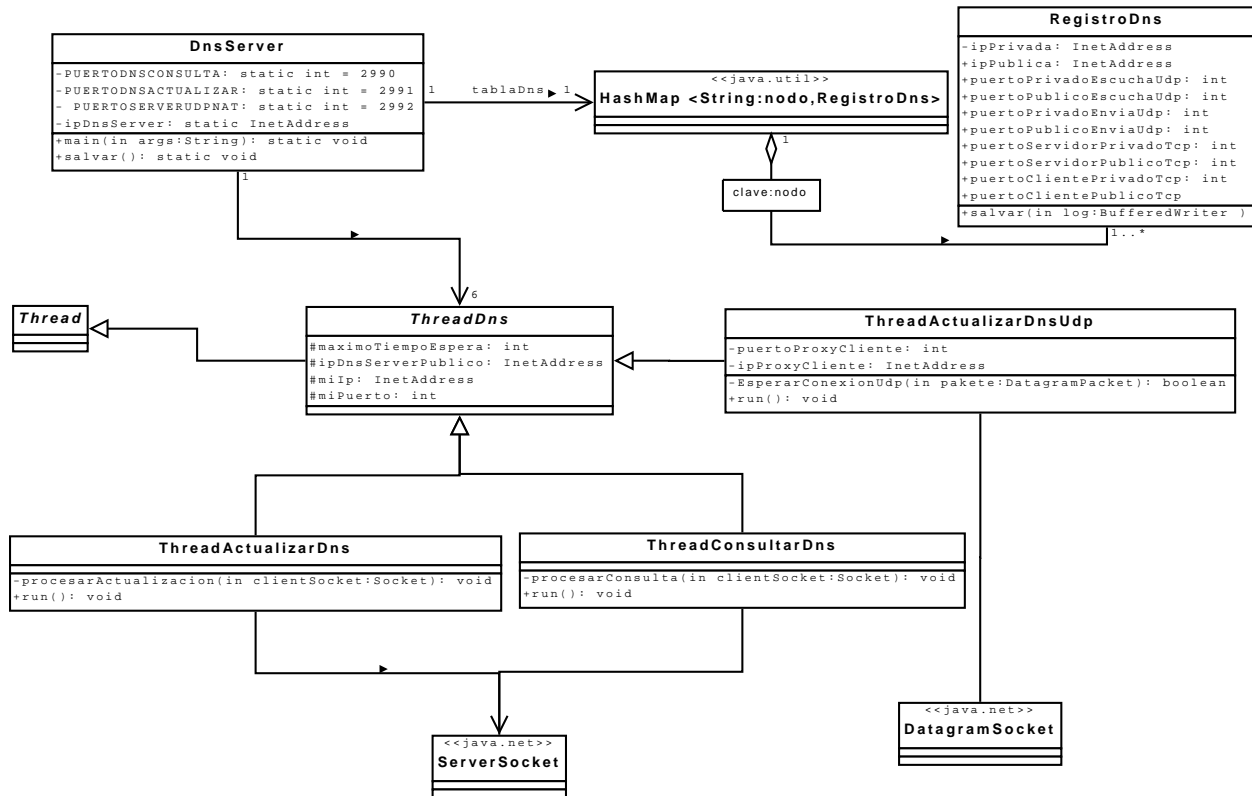


Figura 10.10: Diagrama de clases del paquete dns.

## MetroRed:coordinador

El paquete coordinador es responsable de:

- Leer la agenda de experimentos a realizar y a configuración de cada uno de ellos (tipo de tráfico, topología de la red para el experimento, repeticiones de cada prueba en el experimento, etc.)
- En el tiempo programado coordinar el experimento. Coordinar el experimento implica:
  - Informarle a cada nodo si participa del experimento
  - Configurar a cada nodo que participa del experimento como generador de tráfico, como reenviador o como receptor.
  - Configurar a cada nodo sobre la topología del experimento.
- Luego de coordinado el experimento, cuando todos los nodos ya fueron configurados, informar al nodo que genera tráfico en este experimento que puede comenzar el envío.



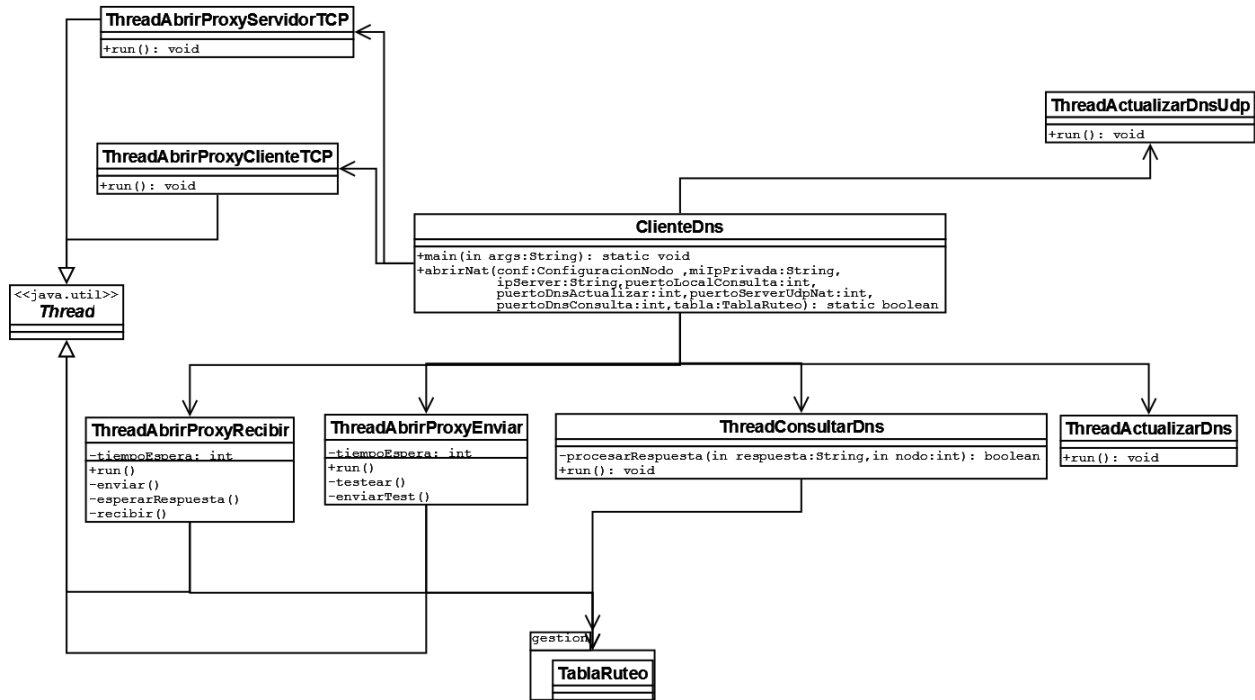


Figura 10.11: Diagrama de clases del paquete clienteDns.

- Esperar a que el generador del experimento le informe que se finalizó de generar el tráfico programado.
- Informarle a todos los nodos el fin del experimento.

En la figura 10.12 se muestra el diagrama de clases de este paquete.

### MetroRed:nodo

El paquete nodo es responsable de:

- Conectarse periódicamente al coordinador para ver si hay algún experimento para realizar y que involucre a este nodo.
- Si hay un experimento coordinar dicho experimento, que involucra:
  - Registrar la topología del experimento de donde sabrá a quién debe enviarle tráfico y de quién debe recibir en este experimento.
  - Si se lo coordina como generador de tráfico, configurar su generador de tráfico de acuerdo a las necesidades del experimento

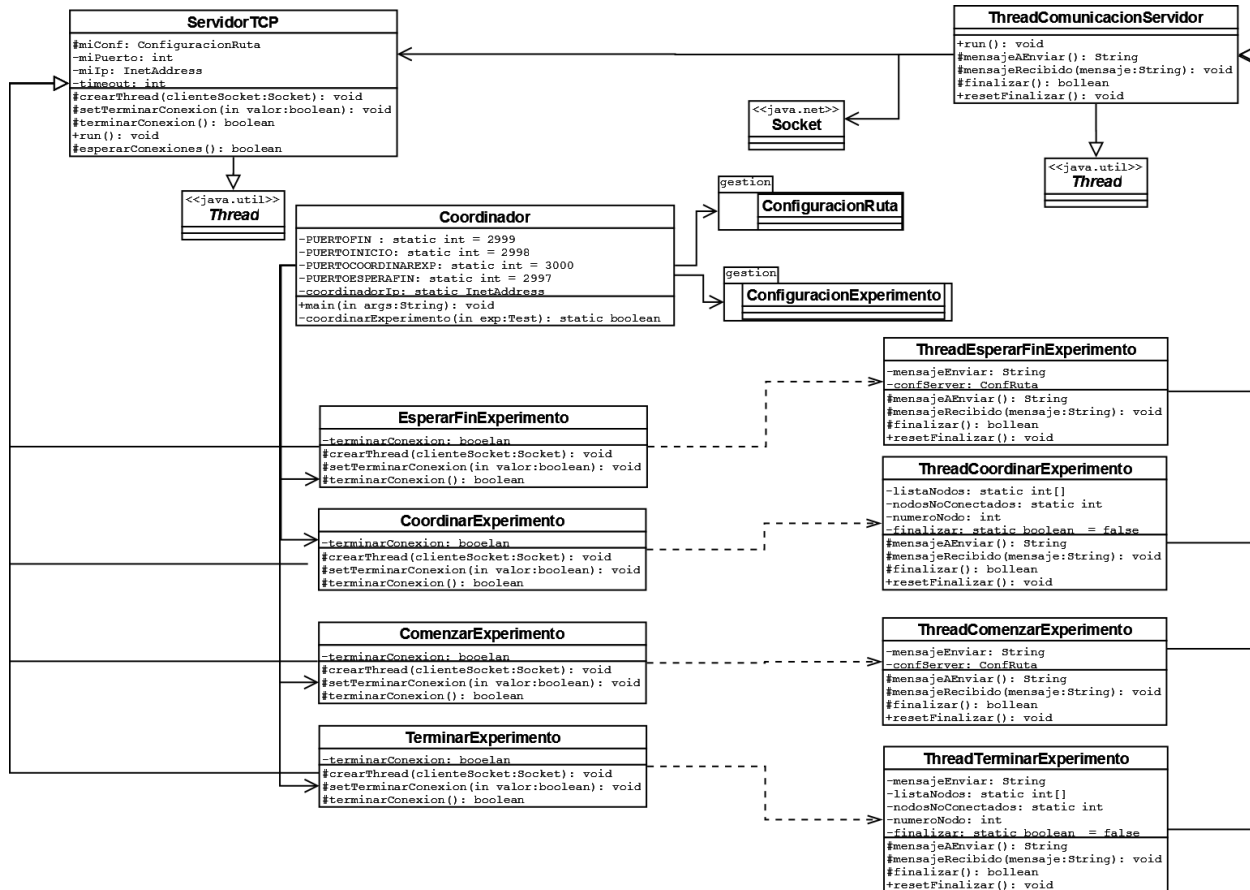


Figura 10.12: Diagrama de clases del paquete coordinador.

- Si va a reenviar paquetes en este experimento, configurar el reenvío y poner a correr el proceso de reenvío.
- Luego de coordinado el experimento, consultar al Dns a través de su clienteDns asociado para obtener ips y puertos de los nodos involucrados en el experimento.
- Luego de coordinado el experimento, si es un nodo generador conectarse al coordinador para que le dé la orden de comenzar a generar tráfico. Cuando recibe esta orden lanzar el proceso que genera tráfico.
- Si es generador, esperar a que se finalice de generar el tráfico configurado en el experimento y cuando esto sucede informarle al coordinador que finalizó.
- Esperar que el coordinador le envíe el fin del experimento. A partir de ese momento vuelve a consultar al coordinador por nuevos experimentos.

En la figura 10.13 se muestra el diagrama de clases de este paquete.

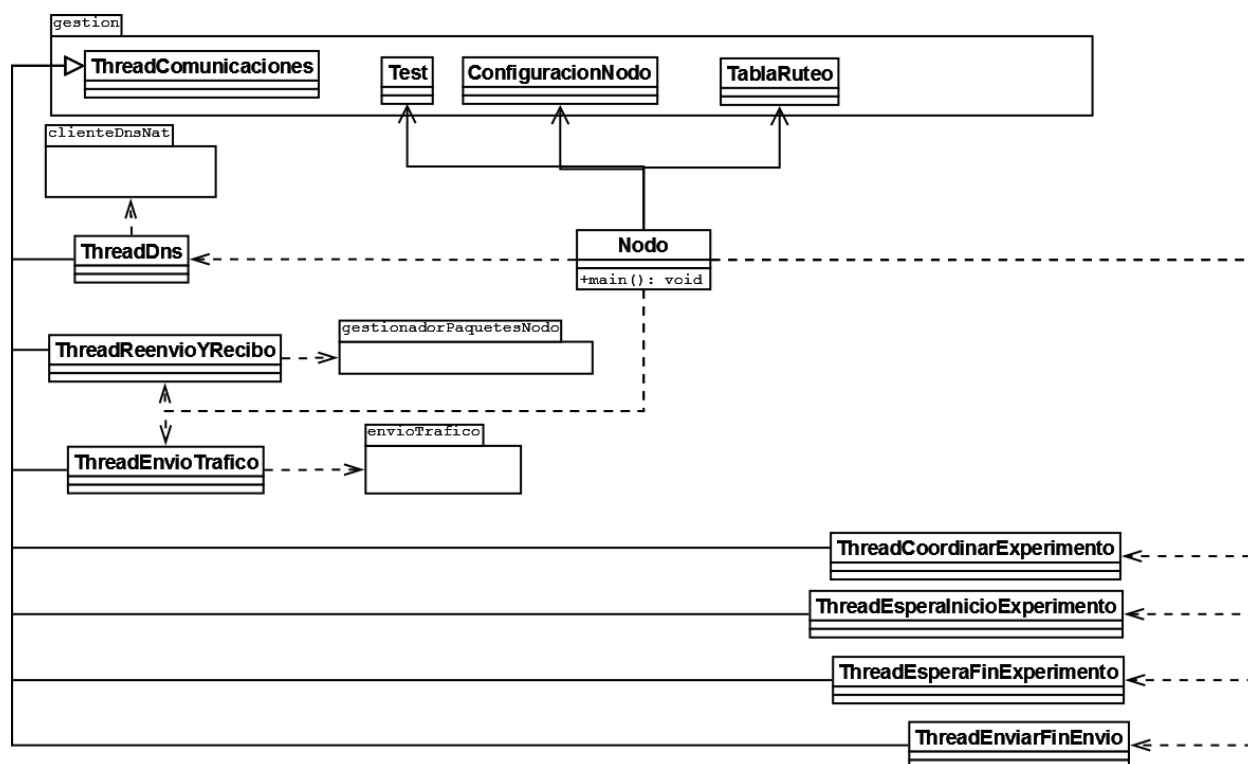


Figura 10.13: Diagrama de clases del paquete nodo.

### MetroRed:generador

Este paquete es responsable de generar tráfico. Contiene a los diferentes generadores de tráfico que heredan de la clase `Generador` que define una interfaz común de los diferentes generadores. En este momento se encuentran implementadas tres generadores: `GeneradorFTP` que envía un archivo vía ftp, `GeneradorPaquetesFijos` que envía paquetes cada tiempo fijo, `GeneradorMedio` que envía tráfico de acuerdo a la especificación de una traza de tráfico definida en un archivo.

En la figura 10.14 se muestra el diagrama de clases de este paquete.

### MetroRed:envioTrafico

Este paquete es responsable de coordinar a los generadores necesarios para enviar el tráfico en el experimento. Es decir, los generadores se pueden combinar de diferentes maneras para generar el tráfico del experimento (por ejemplo, enviar primero paquetes de prueba y luego una traza). Las clases de este paquete combinan los generadores para enviar el tráfico necesario para cada

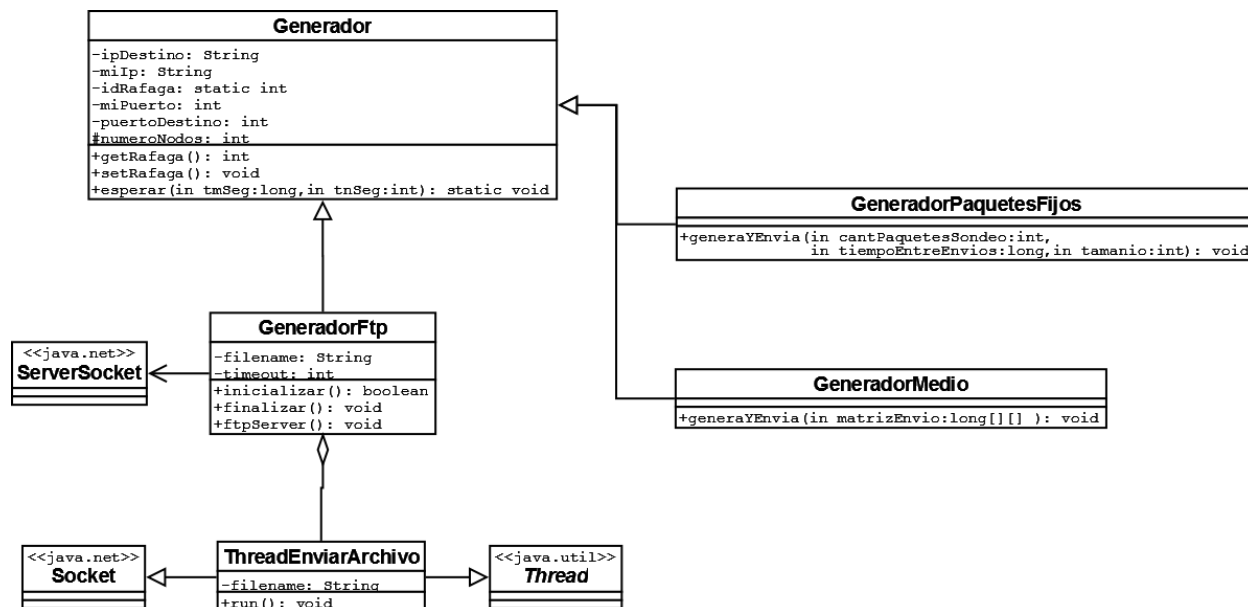


Figura 10.14: Diagrama de clases del paquete generador.

experimento. Todas las clases heredan de la clase EnvioTrafico que define la interfaz de las clases responsables de enviar tráfico.

En la figura 10.15 se muestra el diagrama de clases de este paquete.

### MetroRed:gestionadoresPaquetesNodo

Este paquete es responsable de realizar el reenvío o la recepción del tráfico. Si se envían paquetes de prueba o una traza de tráfico UDP y no es receptor los reenvía al próximo nodo en la topología. Si es receptor recibe los paquetes UDP y guarda su información en un archivo (si reenvía también se puede configurar que guarde la información de los paquetes hasta ese nodo). Si es receptor TCP, baja un archivo desde el servidor TCP del experimento. Además el paquete contiene clases que pueden hacer un volcado de todos los paquetes que pasan por una interfaz del nodo.

En la figura 10.16 se muestra el diagrama de clases de este paquete.

### MetroRed:gestion

El paquete gestion tiene diversas clases auxiliares o que desarrollan funcionalidades comunes a muchos otros paquetes como por ejemplo: manejo de direcciones ip, manejo de comunicaciones a través de sockets, manejo de matrices, etc..

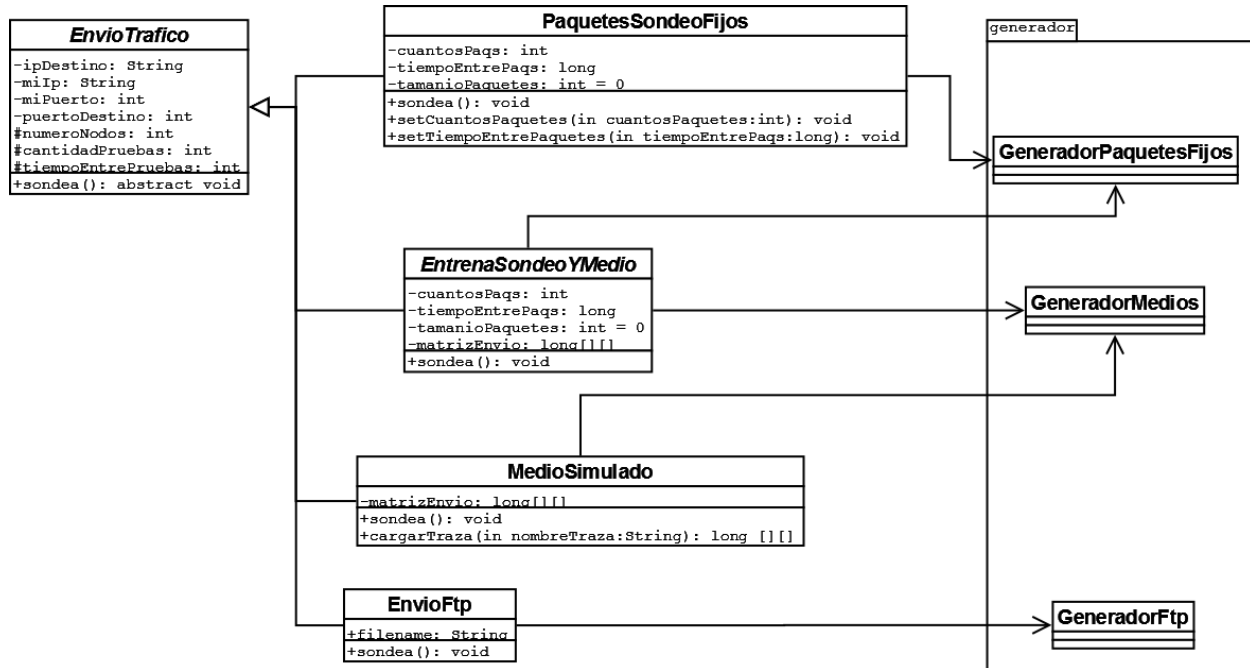


Figura 10.15: Diagrama de clases del paquete envioTrafico.

### EstimacionQoS:procesarRegistros

Este paquete toma la información generada en el experimento y procesa los registros: implementa el estimador de la cola, calcula los diferentes parámetros de QoS: retardo, jitter pérdidas, etc.

### EstimacionQoS:estimarNadaraya

Este paquete es responsable de recibir un vector de puntos  $(X_i, Y_i)$  de entrenamiento y un nuevo punto  $X$  y realizar la estimación de la  $\hat{Y}$  usando el estimador de Nadaraya-Watson. Permite también definir diferentes distancias para las  $X_i$ .

### EstimacionQoS:estimarSVM

Este paquete es responsable de recibir un vector de puntos  $(X_i, Y_i)$  de entrenamiento y un nuevo punto  $X$  y realizar la estimación de la  $\hat{Y}$  usando el estimador de Support Vector Machines. Para este fin utiliza la biblioteca libsvm [34].

### EstimacionQoS:estadística

Este paquete es responsable de calcular diferentes funciones estadísticas sobre un vector de datos: media, mediana, varianza, función distribución, cuantiles, etc..

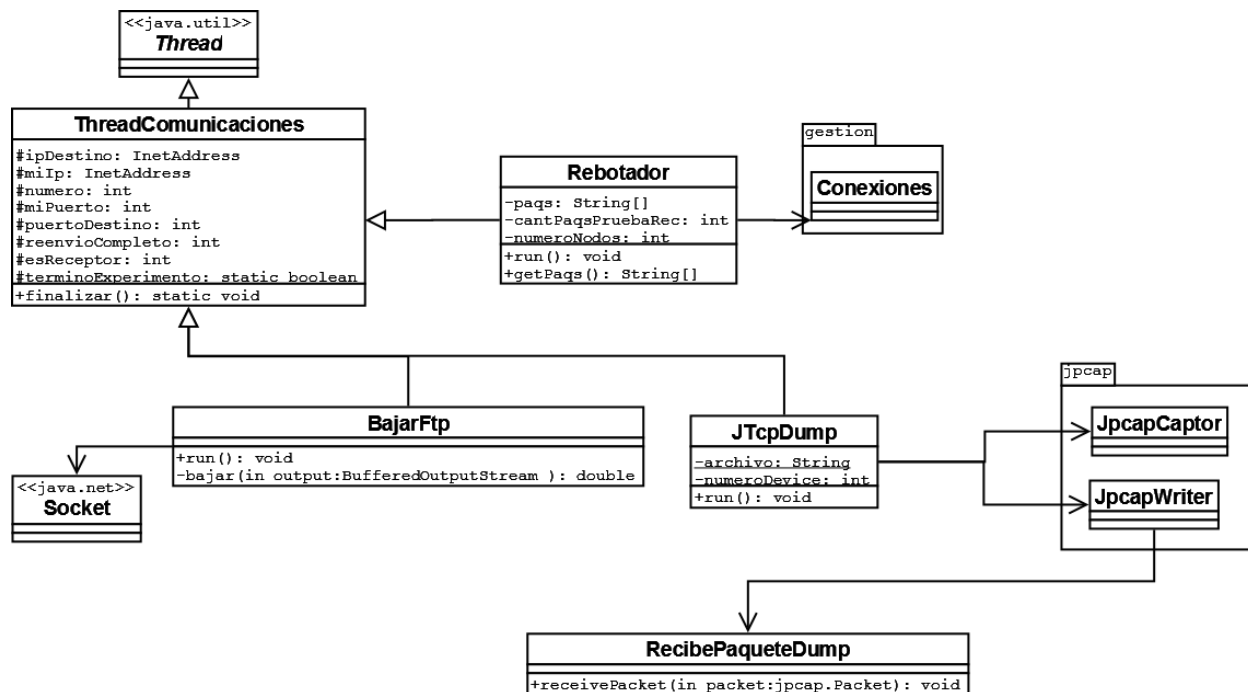


Figura 10.16: Diagrama de clases del paquete gestionadoresPaqueteNodo.

## 10.4. Conclusiones.

Se desarrollaron durante la tesis dos herramientas de medición activa. Con la experiencia del uso de MetroNet, se desarrolló MetroRed. Ambas herramientas han sido probadas y en el capítulo siguiente se muestran resultados obtenidos con MetroNet y MetroRed. La arquitectura de MetroRed (como arquitectura de mediciones overlay distribuida), es una arquitectura novedosa y que puede aplicarse a otras redes overlay que puedan necesitar desplegar una plataforma de mediciones.

Una limitación menor para el uso práctico es la necesidad de contar con una máquina virtual Java en el equipo del cliente. Dada la facilidad de obtener este software sin costo en Internet la alternativa de manejar diferentes versiones del software para la plataforma del usuario pareció una complicación mayor que la de obtener la máquina virtual Java, muchas veces ya existente en las instalaciones de las máquinas cliente.

Aunque el trabajo en un área dinámica como la estimación de calidad de servicio en redes IP hace esperar la aparición de metodologías nuevas y mejores, el diseño del software permite una adaptación fácil, concentrando los cambios en puntos bien específicos, no afectando el resto del sistema.

# Capítulo 11

## Resultados experimentales.

### 11.1. Introducción

En este capítulo se presentan resultados de los métodos de estimación de la QoS de extremo a extremo que percibe una aplicación utilizando tráfico de prueba. Los experimentos se realizaron con el software MetroNet y MetroRed descrito antes. No ha sido posible probar las herramientas de estimación en una red de servicio premium con control de admisión, por no disponer de esta infraestructura para las pruebas. Por este motivo, se buscaron diversos ambientes sobre redes operativas reales, donde en varios de estos ambientes las condiciones probablemente sean más exigentes que en una infraestructura con sólo cierto tipo de servicios y con control de admisión. Por otra parte en las pruebas se buscaron los límites de estos sistemas para aplicaciones de monitoreo en redes menos controladas.

Se presentan experimentos realizados en tres ambientes diferentes. En la sección 11.2, se presentan medidas realizadas entre un servidor ubicado en la Facultad de Ingeniería y un usuario conectado a través de un enlace ADSL. En la sección 11.3 se presentan los resultados de experimentos realizados entre un servidor ubicado en la Facultad de Ingeniería y un servidor ubicado en Canadá. Por último, en la sección 11.4 se presentan medidas realizadas a través de un enlace celular.

### 11.2. Pruebas ADSL

Para verificar el funcionamiento del software y los algoritmos implementados se realizaron diversos ensayos. Se exponen dos casos, ambos realizados desde una máquina de usuario con conexión doméstica habitual ADSL. El parámetro de calidad estimado fue, en ambos casos, el retardo de un video codificado a 382 kbps. Las gráficas muestran la comparación entre la predicción y la realidad.

Los resultados expuestos en la figura 11.1 corresponden a tráfico generado como respuesta a operaciones directas de un usuario experimentador, registrando las diferentes situaciones de demanda de tráfico. La estimación se realiza usando como muestra de aprendizaje todos los puntos anteriores de la muestra. Por lo tanto, los primeros puntos de la gráfica es razonable que brinden estimaciones imprecisas, y la precisión debería ir aumentando a medida que se obtienen más puntos para utilizar

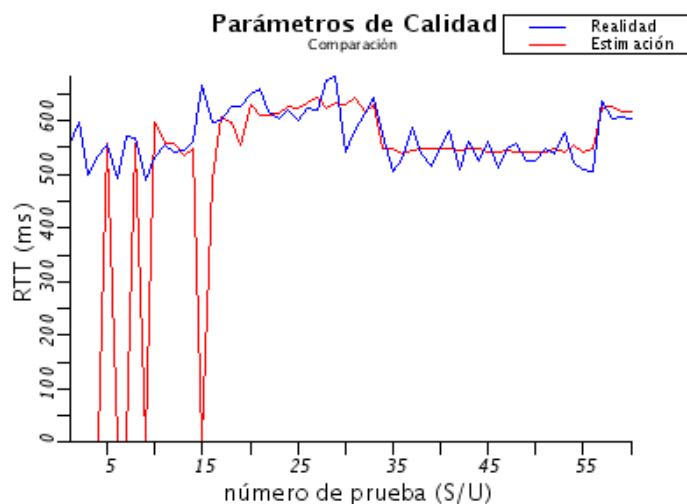


Figura 11.1: Tráfico peer-to-peer.

en la muestra de aprendizaje. Las primeras estimaciones en 0 en dicha gráfica indican la no disponibilidad de datos para poder estimar. Entre las pruebas 10 y 14, cuando la estimación parece haberse estabilizado y comienza a seguir de cerca el tráfico real se arranca una conexión peer to peer, lo cual produce un cambio abrupto en las condiciones que obliga a recomenzar su estimación sin datos históricos. El cambio abrupto de situación hace recomenzar el proceso de estimación sin historia, lo cual se visualiza en la caída a 0 de la prueba 15. Culminado el nuevo período de aprendizaje la estimación vuelve a seguir de cerca la realidad. En la prueba 35 se desconecta la aplicación peer to peer, volviendo a la situación anterior, sobre la cual ya se contaba con información, por lo cual la estimación sigue siendo buena. Finalmente, en la prueba 55, se reinicia la aplicación peer to peer, reiterando la situación anterior, ahora conocida y sobre la cual existen los datos históricos recogidos al principio. Este ciclo de cambios muestra la utilización efectiva de los datos históricos, así como la detección certera de la similitud de situaciones.

El próximo caso, corresponde al monitoreo de un enlace ADSL sin intervención de un usuario. Se tiene una muestra de 50 observaciones correspondientes al tiempo entre paquetes para las ráfagas de paquetes de prueba de donde se obtiene la distribución empírica, junto con el retardo para la secuencia de video. En la figura 11.2 se muestran 30 datos  $X$ .

Estos datos se usarán al mismo tiempo como muestra de entrenamiento y de validación. Para cada observación se estima  $Y_i$  mediante  $\phi_n(X_i)$  donde  $\phi_n$  es la estimación obtenida con todos los demás datos. La estimación y el error relativo se presentan en las figuras 11.3 y 11.4.



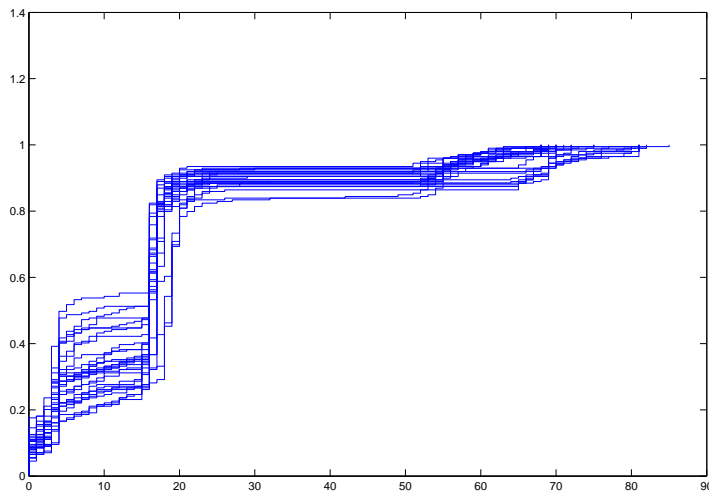


Figura 11.2: Las variables  $X$  son las funciones de distribución empíricas. Se presentan 30 observaciones  $X_1, \dots, X_{30}$ .

### 11.3. Prueba a través de múltiples dominios

En este último caso se buscó probar algunos límites de los métodos de estimación de la calidad de servicio de extremo a extremo. Para esto se instalaron dos nodos MetroRed uno en la Facultad de ingeniería ([arbolita.fing.edu.uy](http://arbolita.fing.edu.uy)) y otro ubicado en Canadá (69.77.187.105). Se instaló el coordinador y el servicio dns de MetroRed en el equipo 69.77.187.105. Se enviaron videos codificados a 382 kbps y paquetes de prueba que representaban una tasa de 42 kbps. Por lo tanto, la carga para la red de los paquetes de prueba es aproximadamente 10 veces menos que la carga que representan los videos.

El camino entre el servidor 69.77.187.105 y [arbolita.fing.edu.uy](http://arbolita.fing.edu.uy) pasa por 25 enlaces como se muestra en el cuadro 11.1.

En el cuadro 11.1 se puede observar además que hay un nodo (el 13) que balancea carga. Se observa también que el retardo más importante se genera en el enlace internacional ANTEL-Sprint (del orden de 150 ms). Los experimentos se desarrollaron durante aproximadamente 10 días. Al pasar por tantos enlaces la conexión en el correr de esos días hubo algunos cambios de rutas. La ruta presentada en la tabla anterior es la que se observó durante casi todo el tiempo que duró el experimento. La mayoría de los cambios de ruta observados durante los experimentos duraban del orden de pocas horas y se volvía a la misma ruta original. Hubo una excepción que se analizará más adelante donde el cambio de ruta fue de aproximadamente un día. Durante el experimento se realizaron pruebas enviando los paquetes de prueba y el video para estimar los parámetros de QoS cada un tiempo entre 15 y 30 minutos. Hay que observar además que los

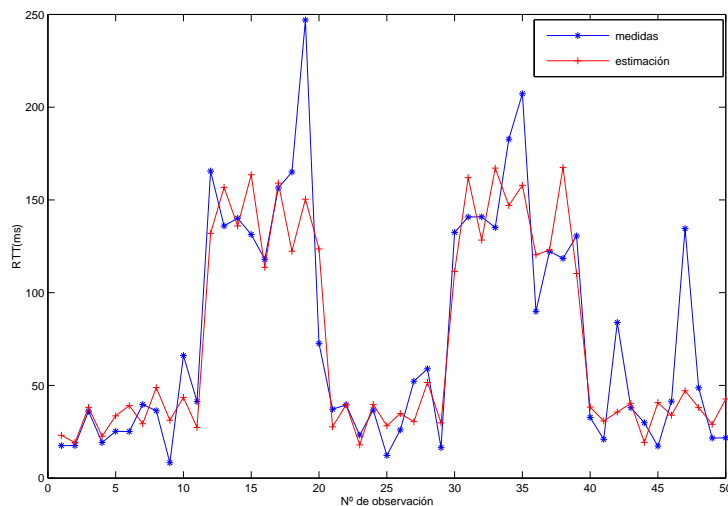


Figura 11.3: RRT estimado y real

enlaces de acceso en este camino son de 4Mbps y 10Mbps. Los demás enlaces son todos de mayor capacidad aún. Esto presenta una dificultad adicional, por la precisión por las pequeñas diferencias de tiempo involucradas.

Se consideraron las medidas de 7 días como muestra de entrenamiento y se estimaron dos días aproximadamente con dicha muestra de entrenamiento.

En primer lugar, se realizó una regresión funcional utilizando como variable  $X$  la distribución empírica de la cola y como variable  $Y$  a estimar el retardo medio. Se obtuvo un error cuadrático medio de 8.6 ms sobre la muestra de dos días de validación. Posteriormente se realizó una regresión ahora con SVM y utilizando como variable  $X$  diferentes características de la distribución de la cola individualmente. En el cuadro 11.2 se lista el error cuadrático medio (MSE) obtenido con cada característica individual.

A partir de esta información se evaluaron algunas combinaciones de características descartando aquellas que a priori brindan muy poca información. Se encontró que utilizando el valor medio de la cola, el percentil 70% y el porcentaje de cola vacía brindaban un MSE similar al que se obtiene con la regresión funcional utilizando la distribución empírica de la cola.

En la figura 11.5 se muestra el valor real y el estimado utilizando las tres características mencionadas en el párrafo anterior. Como se puede apreciar la estimación es razonable teniendo en cuenta las consideraciones realizadas sobre las características del camino utilizado.

En la figura 11.6 se muestra el valor estimado y el medido del jitter del video, en lugar del retardo. Se utilizó la misma muestra de entrenamiento y de validación que en el caso anterior pero en este caso la variable  $Y$  a estimar fue el jitter. En este caso, para obtener un error cuadrático

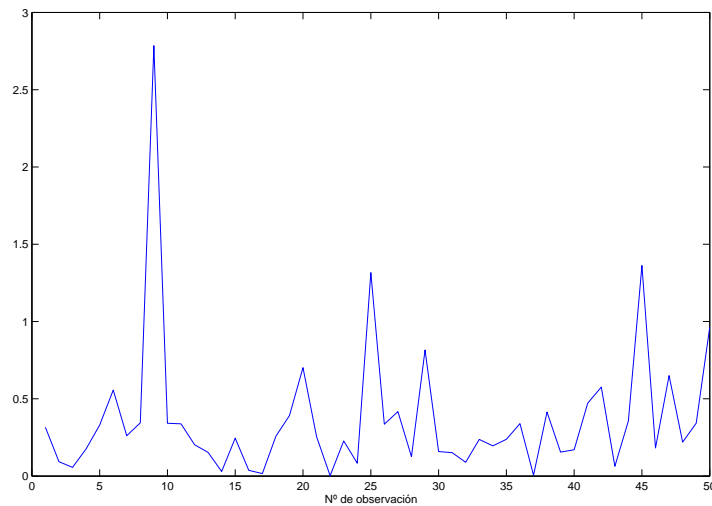


Figura 11.4: Error relativo.

medio similar al que se obtiene usando la distribución empírica de la cola, se debieron considerar las siguientes 9 características: valor medio de la cola, varianza, percentiles 50,60,70, 80, 90, 95 % y el porcentaje de tiempo de cola vacía.

En la figura 11.7 se muestra la estimación de la tasa de pérdidas de paquetes en el camino. Las muestras utilizadas son las mismas que en los casos anteriores. En este caso se probó estimar las pérdidas con tres estimadores  $X$  diferentes.

Caso 1. Se estimaron las pérdidas del video utilizando como variable  $X$  la distribución empírica de la cola.

Caso 2. Se estimaron las pérdidas del video pero utilizando como variable  $X$  las pérdidas que experimentaron los paquetes de prueba.

Caso 3. Se estimaron las pérdidas del video usando como variable  $X$  una combinación de los anteriores: valor medio de la cola, varianza, percentiles 50,60,70, 80, 90, 95 %, el porcentaje de tiempo de cola vacía y pérdidas de los paquetes de prueba.

El error cuadrático medio en estos tres casos se muestra en el cuadro 11.3:

Por último, se muestra en la figura 11.8 la estimación del retardo medio hecha con la misma muestra de entrenamiento pero la muestra de validación corresponde a otro día de las pruebas. Es interesante observar que al principio de la gráfica y al final la estimación parece adecuada. Sin embargo en el medio si bien aproximadamente “sigue la forma” hay un desfase aproximadamente constante de 5 ms aproximadamente. Analizados los datos de ese día se observa que la ruta cambió. En el cuadro 11.4 se muestran los saltos que cambiaron.

número salto	interfaz	ip	interfaz 2
1	69-77-187-1.onx.com	(69.77.187.1)	
2	192.168.240.1	(192.168.240.1)	
3	69.77.183.13	(69.77.183.13)	
4	69.77.175.90	(69.77.175.90)	
5	69.77.175.94	(69.77.175.94)	
6	69.77.175.66	(69.77.175.66)	
7	69.77.175.38	(69.77.175.38)	
8	142.47.135.1	(142.47.135.1)	
9	142.46.0.14	(142.46.0.14)	
10	ge-5-2-110.hsa2.Detroit1.Level3.net	(64.152.144.1)	
11	ae-8-8.ebr2.Chicago1.Level3.net	(4.69.133.242)	
12	ae-2-54.edge3.Chicago3.Level3.net	(4.68.101.116)	
13	sl-st20-chi-5-0.sprintlink.net	(144.232.19.173)	(144.232.8.113)
14	sl-crs2-chi-0-11-5-0.sprintlink.net	(144.232.8.218)	
15	sl-crs2-chi-0-11-3-0.sprintlink.net	(144.232.20.53)	
16	sl-crs1-mia-0-11-2-0.sprintlink.net	(144.232.18.217)	
17	sl-bb20-mia-13-0-0.sprintlink.net	(144.232.2.253)	
18	sl-st21-mia-2-0.sprintlink.net	(144.232.9.199)	
19	sl-antel1-1-0.sprintlink.net	(144.223.245.162)	
20	ibb2uni1-p1.antel.net.uy	(200.40.22.37)	
21	ibb2cen1-1-3.antel.net.uy	(200.40.16.89)	
22	iem2cen1-0-1.antel.net.uy	(200.40.17.50)	
23	seciu-ibgp.adinet.com.uy	(200.40.160.9)	
24	r3.rau.edu.uy	(164.73.128.129)	
25	eth-fing.rau.edu.uy	(164.73.253.34)	

Cuadro 11.1: Traceroute

Como se puede ver a partir del salto 9, en lugar de salir por Detroit y Chicago hacia Miami, se va por Montreal y Nueva York. Este cambio de ruta probablemente genere un cambio en la latencia total y hace que se desfase el retardo, pero aparentemente las colas en esta zona (backbone) no afectan mayormente por lo que se sigue bien el andamiaje del retardo (salvo algún pico) si se agrega este desfase.

Por último se debe observar en las diferentes gráficas vistas que hay algunos valores altos (picos) de las diferentes variables a estimar que en algunos casos se subestiman. Aquí hay dos aspectos a tomar en cuenta. Por un lado los picos son eventos “raros” por lo que se necesita mucho más datos para aprenderlos correctamente. En segundo lugar se asume que hay estacionariedad durante cada prueba, pero pueden suceder eventos esporádicos que se den sólo sobre la  $X$  o sobre la  $Y$  y obviamente en estos casos existirán errores en la estimación de estos puntos. Para disminuir el impacto de este problema lo que se hizo fue en lugar de enviar 40 segundos de video y 40 segundos de paquetes de prueba, se intercalan video y paquetes de prueba enviando 5 segundos de  $c/u$  de forma de que si hay un evento transitorio sea más probable que afecte a la  $X$  y a la  $Y$  y no a ambas.

Característica	MSE
valor medio de la cola	10.8
varianza de la cola	12.8
percentil 10 %	36
percentil 20 %	36
percentil 30 %	25
percentil 40 %	14
percentil 50 %	12.8
percentil 60 %	11.8
percentil 70 %	11.8
percentil 80 %	12.3
percentil 90 %	13.6
percentil 95 %	13
percentil 99 %	26
porcentaje de cola vacía	16
pérdidas de paquetes de prueba	23

Cuadro 11.2:

Caso	MSE
1	$2 \cdot 10^{-4}$
2	$1,6 \cdot 10^{-4}$
3	$1,2 \cdot 10^{-4}$

Cuadro 11.3:

## 11.4. Prueba a través de un enlace celular

En este caso se utilizó un enlace GPRS/EDGE a través un PC y modem EDGE ubicado en una radio base a 300 km de Montevideo. El servidor desde donde bajar las secuencias de video fue arbolita.fing.edu.uy, ubicado en la Facultad de Ingeniería, UDELAR. Se enviaron videos codificados a 96 kbps dado que la tasa de bajada máxima en un enlace EDGE es del orden de 240 kbps.

Se hicieron 65 pruebas de bajada de video y paquetes de prueba. En este caso los valores del retardo observado son importantes y presentan mucha variabilidad. Además, se tienen pocas muestras como para partir la muestra de entrenamiento, seleccionar los parámetros de SVM ( $\gamma, C$ ) y validar el resultado. Cuando se tienen pocas muestras, una técnica que se utiliza habitualmente es “one left out”. Esta técnica se utiliza tanto para validación como para selección de modelos. Para seleccionar el modelo con las muestras de datos obtenidos en GPRS/EDGE se podría utilizar así: para cada pareja ( $\gamma, C$ ) fija, se estima cada punto con el resto la muestra (con los 64 valores restantes). Se calcula el error cuadrático medio de los 65 puntos de la muestra para cada valor fijo de parámetros. Por último, se elige el par de parámetros que minimiza el error cuadrático medio. Cuando se utilizó este procedimiento con los datos obtenidos de la red GPRS/EDGE, se se seleccionó el modelo y luego con ese modelo se validó. En la validación, se observó que la estimación no era lo suficientemente buena. Analizado este resultado, se concluyó que la existencia de datos no-estacionarios en la muestra podía ser la causante de este problema. Si bien la no-estacionariedad también se daba en el caso de la mediciones realizadas a través de varios dominios en Internet, como

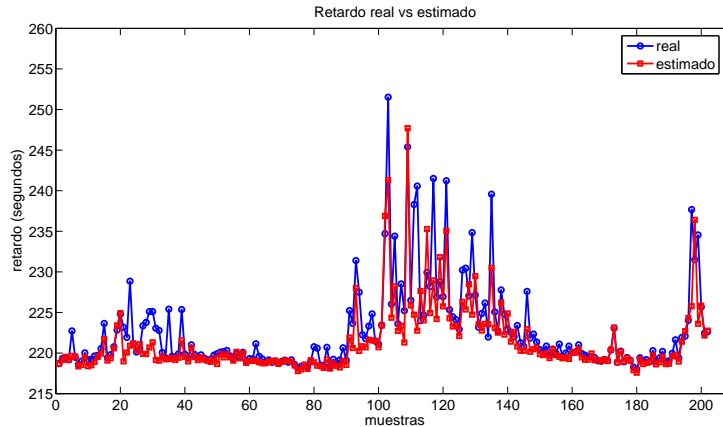


Figura 11.5: Valor medido y estimado del retardo medio de video

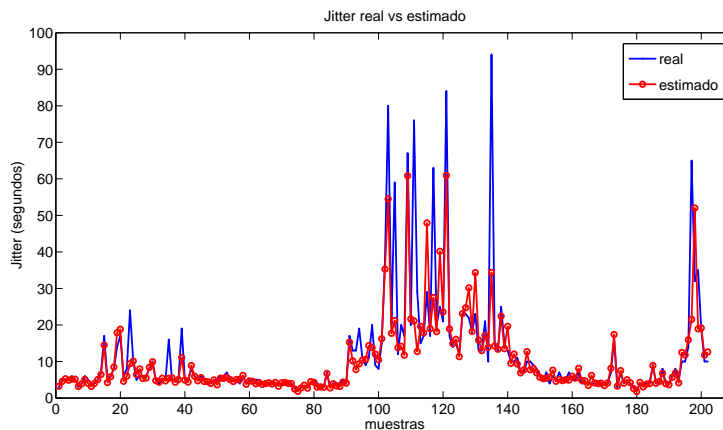


Figura 11.6: Valor medido y estimado del jitter del video

la muestra en ese caso era lo suficientemente grande este problema era menos notorio. El problema se refleja en que la pareja  $(\gamma, C)$  que se obtiene al ajustar el modelo, aproxima bien los puntos donde se dan los picos del retardo o las pérdidas o aproxima bien los valles pero no se obtiene una buena aproximación en ambas zonas.

Este problema ya fue observado y analizado por diversos autores [85, 88, 113, 120] al aplicar SVM a otros tipos de datos no estacionarios, por ejemplo para datos biológicos. En los trabajos anteriores se buscan diferentes procedimientos para solucionar este problema. La idea principal, común a varias propuestas, es que en lugar de buscar un único núcleo o conjunto de parámetros diferentes se busca un conjunto núcleos o de parámetros para las diferentes componentes estacionarias de la señal (por ejemplo un conjunto de parámetros diferentes, para diferentes escalas de tiempo).

En la tesis se siguió la idea anterior, pero de una forma particular que se adapta bien al problema específico.

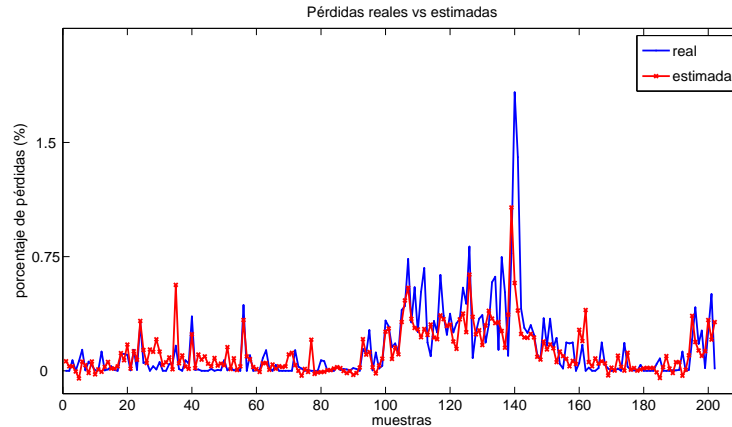


Figura 11.7: Valor medido y estimado de la tasa de pérdidas de paquetes de video

número salto	interfaz	ip	interfaz 2
9	142.46.128.90	(142.46.128.90)	
10	if-1-150.mcore3.MTT-Montreal.as6453.net	(216.6.111.13)	
11	if-13-0.mcore4.NQT-NewYork.as6453.net	(216.6.87.21)	
12	if-5-0.mcore3.NYY-NewYork.as6453.net	(216.6.87.54)	
13	if-12-0-0-723.core4.AEQ-Ashburn.as6453.net	(216.6.42.61)	
14	sl-st20-ash-14-0-1.sprintlink.net	(144.223.246.105)	
15	sl-bb21-dc-10-0-0.sprintlink.net	(144.232.20.151)	
16	sl-bb21-mia-6-0-0.sprintlink.net	(144.232.9.26)	
17	sl-st21-mia-3-0-0.sprintlink.net	(144.232.2.240)	
18	sl-st21-mia-2-0.sprintlink.net	(144.232.9.199)	

Cuadro 11.4: Cambio de ruta

En primer lugar, para seleccionar el modelo, lo que se hizo fue tomar un subconjunto de las 65 muestras (las primeras 30) y dividir dicho subconjunto de acuerdo al valor de las  $Y_i$  en tres clases (una correspondiente a los picos, otra a los valles y otra a la transición entre estos). Ese subconjunto de 30 muestras, se divide en tres clases cortando por niveles de las  $Y$  de forma que se obtengan tres muestras de igual cantidad de elementos (10 muestras cada clase). Sobre cada clase  $j$  se calcula el par de parámetros  $(\gamma_j, C_j)$  que minimiza el error cuadrático medio de la clase correspondiente, utilizando “one left out”. Una vez dividida las clases según las  $Y_i$ , además se calcula el baricentro de las  $X_i$  de cada clase  $X_j$ .

Para validar, se toman los restantes 35 puntos que no se usaron para seleccionar el modelo de cada clase. Para cada punto  $X_k$  de la muestra de validación, se determina la clase  $j_1$  de la que este punto se encuentra a menor distancia de su baricentro  $X_j$ . Una vez identificada la clase  $j_1$ , se seleccionan los parámetros de la clase  $\gamma_{j_1}, C_{j_1}$ . Por último, se estima el  $Y_k$  con dichos parámetros y utilizando “one left out” sobre la muestra de 65 puntos.

En las figuras 11.9, 11.10 se muestra el porcentaje de pérdidas de paquetes y el retardo respectivamente para los últimos 35 puntos de la muestra. En ambas figuras se muestra el valor medido y

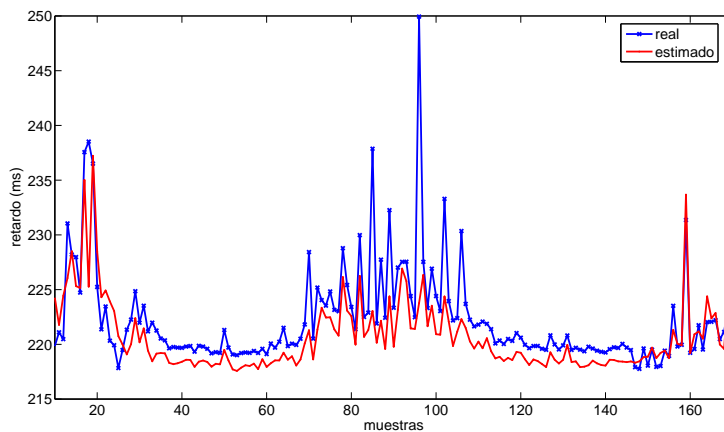


Figura 11.8: Cambio de ruta

el estimado. Como se puede ver la aproximación es razonablemente buena para la variabilidad que presentan los datos.

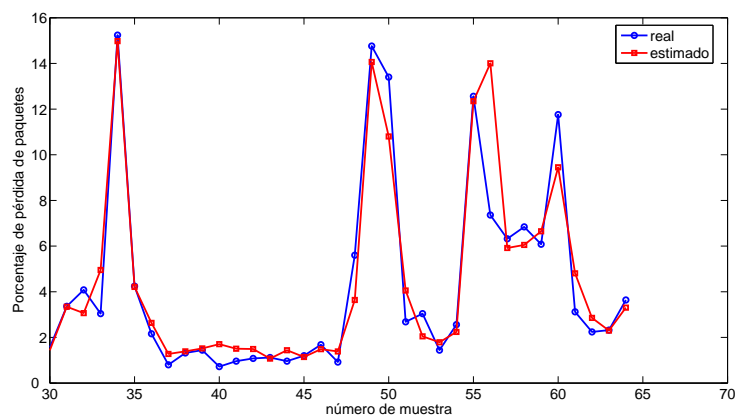


Figura 11.9: Porcentaje de pérdidas de paquetes de video sobre un acceso GPRS/EDGE

## 11.5. Conclusiones

En este capítulo se mostraron resultados sobre tres casos diferentes de redes operacionales públicas, y con tráfico real. En el primer caso se mostraron resultados obtenidos utilizando un acceso ADSL contra un servidor ubicado en la Facultad de Ingeniería. En estos casos con algunas pocas decenas de muestras de entrenamiento se obtuvieron buenas aproximaciones. En segundo lugar se analizaron diferentes parámetros de desempeño sobre una camino en Internet entre un



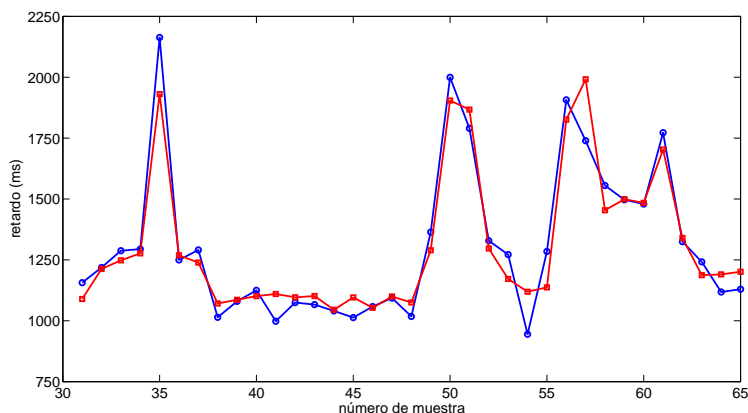


Figura 11.10: Retardo de paquetes de video sobre un acceso GPRS/EDGE

equipo ubicado en la Facultad de Ingeniería y un equipo ubicado en Canadá. En el caso anterior muy probablemente el enlace con encolamiento de paquetes fuera uno sólo y estuviera ubicado en el acceso ADSL. En este caso en cambio, es mucho más difícil predecir dónde y cuántos puntos de enlaces con encolamiento de paquetes puede haber. Además todos los enlaces en este caso tienen capacidades superiores a los 4 Mbps e incluso hay enlaces que balancean carga en el camino. A pesar de estas dificultades se logran aproximaciones razonables pero con muestras de entrenamiento mayores (algunas centenas de puntos). El otro resultado interesante es que esta aproximación se logra con pocas características de la distribución del estimador de la cola. Por último se analiza el caso de un enlace celular. La característica de este caso es que los datos presentan importantes variaciones y además se dispone sólo de 65 muestras. Por este motivo, en este caso quedó en evidencia problemas para ajustar los parámetros de SVM por la falta de estacionariedad de los datos. Se propuso entonces un método para ajustar el modelo SVM con varios conjuntos de parámetros. Los resultados obtenidos con este procedimiento son razonablemente precisos a pesar de los pocos datos obtenidos y de su variabilidad.

Sin duda las estimaciones se pueden mejorar y hay varios aspectos para investigar tanto en el sistema de mediciones como en los algoritmos de estimación, pero dada las condiciones de algunas de las pruebas realizadas, se entiende que las aproximaciones que se logran con el método desarrollado son aceptables.

# Capítulo 12

## Calidad de servicio percibida

### 12.1. Introducción

Si bien el foco de la tesis no está en el área de calidad de servicio percibida (PqoS), sin embargo hay múltiples puntos de contacto entre las técnicas de regresión y estimación vistas en el capítulo anterior y el área de PQoS de fuerte desarrollo actual. A los efectos de completar el panorama referente a la calidad de servicio de aplicaciones multimedia en Internet, en este capítulo se analizará el estado del arte en PQoS, y se evaluará en qué se diferencian y cómo se pueden complementar las técnicas desarrolladas en esta tesis con algunas técnicas de PQoS.

La calidad de servicio percibida por un usuario de un servicio multimedia en Internet, no sólo depende del comportamiento de la red sino también de las características del servicio (tipo de codificación y compresión utilizado, algoritmos de recuperación de errores y problemas, características del contenido, etc.). Un usuario final puede experimentar niveles de calidad de servicio aceptables aún en la presencia de problemas severos en la red.

El análisis de la calidad de servicio percibida por un usuario puede ser analizada por métodos *subjetivos* o por métodos *objetivos*. La figura 12.1 presenta una visión general de las técnicas utilizadas en el área de calidad de servicio percibida.

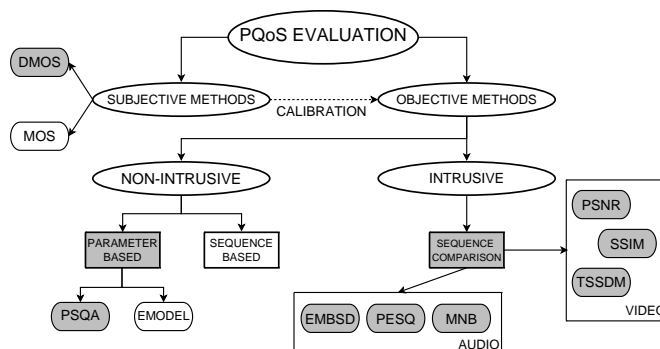


Figura 12.1: Clasificación de las técnicas utilizadas en PQoS.

Las técnicas subjetivas se basan en la opinión media de la evaluación hecha por un grupo de usuarios de un servicio de audio o video. Existen diferentes recomendaciones a los efectos de realizar estos test subjetivos de un grupo de usuarios tanto para el caso de audio [65] como para el caso de video [66, 67]. El problema de las técnicas subjetivas es que implican un costo importante en tiempo al menos para realizar la evaluación por los usuarios y eso lleva a dificultades para automatizar este tipo de métodos.

Por otro lado las técnicas objetivas no dependen de los usuarios directamente. Las técnicas objetivas de evaluación de PQoS pueden clasificarse en *intrusivas* o *no-intrusivas*.

Los métodos intrusivos se basan en comparar dos secuencias: la secuencia original y la secuencia obtenida luego de que la secuencia original es enviada a través de la red. La comparación entre ambas secuencias puede ser hecha tanto analizando las relaciones espacio/temporales de las señales (comparando el error cuadrático medio (MSE), la relación señal a ruido (SNR) o la relación entre la señal de pico y el ruido (PSNR)) o realizando un análisis en el dominio *perceptual*. En este último caso se utilizan además modelos para los sentidos con que percibe la señal un humano. Ejemplos de métodos en el dominio perceptual para señales de audio son “the perceptual speech quality measure (PSQM)” [68], “the measuring normalizing blocks (MNB)” [149], “the enhanced modified bark spectral distortion (EMBSD)” [160] y “the perceptual evaluation of speech quality (PESQ)” [69, 11]. En el caso de señales de video algunos métodos en el dominio perceptual son: “Structural Similarity Index Measurement (SSIM)” [152, 153, 154], “Video Quality Measurement (VQM)” [150] y “Time/Space Structural Distortion Measurement (TSSDM)” [165]. En [151] se presenta una evaluación de los métodos objetivos de PQoS.

El principal problema de las técnicas objetivas e intrusivas para evaluar PQoS es la necesidad de contar con la secuencia original, lo cual en varios escenarios de aplicaciones en redes puede constituir una dificultad importante. Estas técnicas aplicadas al caso de video además consumen muchos recursos computacionales y tiempo por la complejidad involucrada en los cálculos de algunas de estas técnicas.

Por el contrario las técnicas no-intrusivas no requieren de la señal original para su evaluación. Esto permite que puedan ser usadas en aplicaciones de tiempo real. Estas técnicas pueden ser clasificadas en técnicas *basadas en secuencias* o técnicas *basadas en parámetros*. En el caso de las técnicas basadas en secuencias no se requiere la señal original y se basan en aplicar ciertos algoritmos a la señal distorsionada. Estos métodos son referidos en la literatura como de tipo “null reference”. En el caso de las técnicas basadas en parámetros, diferentes características tanto de la red como de la secuencia son tomadas como entradas al algoritmo. La idea es construir un modelo que haga un mapeo entre la PQoS como es percibida por un usuario final y un subconjunto relevante del conjunto de características mencionados. Ejemplos de estas características son: la tasa de pérdidas, el retardo, el jitter, tipo de codificación, naturaleza del contenido (por ejemplo cantidad de movimiento de la secuencia de video), tasa de bits, etc.. El modelo de la ITU E-Model [63] y el modelo de “pseudo subjective quality assessment (PSQA)” [102, 103, 130] son ejemplos de técnicas de este tipo. El E-Model es un conjunto de fórmulas originalmente diseñadas para planificación de redes telefónicas, pero que son utilizadas para redes IP aunque diversos resultados muestran que no brindan una buena aproximación a la estimación de la calidad de servicio percibida [61]. El modelo PSQA utiliza técnicas de aprendizaje estadístico (redes neuronales aleatorias [60]) a los efectos de *aprender* la relación entre las características de la red y el video con la calidad percibida

por los usuarios. La técnica PSQA ha mostrado resultados interesantes en el campo de la calidad de servicio percibida [102, 103]. La principal debilidad de los métodos basados en parámetros es que tienen una fuerte dependencia de los resultados de los test subjetivos que es necesario realizar con los usuarios a los efectos de entrenar el algoritmo de aprendizaje.

En la sección 12.2 se presentan brevemente algunas técnicas de evaluación objetiva de la PQoS y en la sección 12.3 se comparan estas técnicas. La comparación fue realizada utilizando un software desarrollado en un proyecto de grado de Ingeniería Eléctrica. El proyecto fue realizado por P.Casas, D. Guerra e I. Irigaray bajo mi tutoría [31].

## 12.2. Métodos para evaluar la PQoS

### 12.2.1. Evaluación subjetiva

En este tipo de test, un grupo de personas clasifican diferentes secuencias distorsionadas (de audio o video). Existen dos tipos de tests subjetivos dependiendo de si la secuencia original es incluida a o no en la evaluación. Cuando no hay secuencia de referencia las personas sólo clasifican las secuencias distorsionadas de acuerdo a escala como la que se muestra en la tabla 12.1(a); La salida de este test se conoce como Mean Opinion Score (MOS). Hay diferentes test de este tipo para audio y video. Cuando la secuencia de referencia es incluida en el test, las personas comparan la secuencia original con la distorsionada y califican el grado de degradación percibida de acuerdo a una escala como la de la tabla 12.1(b). La salida de este test se denomina “Degradation Mean Opinion Score” (DMOS). En audio el test se denomina “Degradation Category Rating” (DCR) y en video, “Double Stimulus Impairment Scale” (DSIS).

Puntaje	Calidad de la secuencia	Puntaje	Degradación de la secuencia
5	Excelente	5	Imperceptible
4	Bueno	4	Perceptible, sin molestias
3	Regular	3	Algo molesto
2	Malo	2	Molesto
1	Muy malo	1	Muy molesto

(a) Escala de calidad MOS

(b) Escala de calidad DMOS

Cuadro 12.1: Diferentes escalas de calidad.

Hay muchas variantes sobre los test subjetivos definidos en las recomendaciones de la ITU [65] (audio) y [66, 67] (video).

## 12.2.2. Evaluación objetiva - Métodos intrusivos

### Métodos para audio

Con la herramienta de software se compararán tres métodos que se describen brevemente en esta sección: Enhanced Modified Bark Spectral Distortion (EMBSD), Perceptual Evaluation of Speech Quality (PESQ-ITU P.862), y Measuring Normalizing Blocks (MNB). Estos algoritmos realizan la comparación como se explicó en la introducción en el dominio perceptual. Se consideran en ellos tres conceptos psico-acústicos: *las bandas críticas, la intensidad y el enmascaramiento*. *Las bandas críticas* están basadas en la habilidad de un humano de distinguir entre diferentes tonos. En bajas frecuencias pocos hertz son suficientes para distinguir entre tonos, mientras que en altas frecuencias el umbral se eleva a cientos de hertz. En base a estas bandas de frecuencia el auditorio es modelado como un banco de filtros pasa-banda. *La intensidad* considera la *intensidad percibida* de un sonido. Por ejemplo, una señal sinusoidal de 40 dB a 50 Hz es igualmente percibida (en términos de fuerza) como una señal sinusoidal de 0 dB a 1 KHz. La percepción está relacionada tanto con la intensidad como con la duración de un sonido (el auditorio integra la intensidad sobre una cierta ventana de tiempo). *El enmascaramiento* representa el efecto psico-acústico que ocurre cuando la presencia de un sonido inhibe la percepción de otro. Lo que se hace en el test es modificar el umbral de percepción del auditorio por la presencia de un sonido.

### Métodos para video

Los algoritmos que se consideran para la evaluación de video difieren en qué consideran relevante para la percepción de un usuario.

**Mean Square Error (MSE) y Peak Signal to Noise Ratio (PSNR)** Los algoritmos MSE y PSNR son los más simples para comparar dos secuencias. No tienen en cuenta ninguna característica perceptual, sólo comparan píxeles entre cuadros de las secuencias original y distorsionada de video. Las métricas MSE y PSNR son definidos de la siguiente forma:

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (12.1)$$

$$PSNR = 10 \log_{10} \left( \frac{L^2}{MSE} \right) \quad (12.2)$$

donde  $n$  es el número de píxeles en la imagen o video,  $x_i$  y  $y_i$  son el  $i$ -ésimo píxel en las imágenes original y distorsionada respectivamente, y  $L$  es el rango de valores posibles para un píxel. Estos métodos son simples pero tienen poca correlación con los test subjetivos como se muestra en la figura 12.2. En las figuras 12.2(a) y 12.2(b), la imagen original (la de la izquierda) es comparada contra la imagen distorsionada (la de la derecha). Ambos grupos de imágenes tienen aproximadamente el mismo valor de PSNR pero las de la figura(12.2(a)) tienen diferencias poco perceptibles, mientras que las de la figura (12.2(b)) tienen diferencias muy evidentes.



(a) Mandril, PSNR = 159



(b) Morrón, PSNR = 160

Figura 12.2: PSNR como medida de las diferencias percibidas.

**“Time/Space Structural Distortion Measurement” (TSSDM)** El objetivo es medir cambios en la actividad espacial, considerando ciertas regiones espacio-temporales (ST) de los videos original y distorsionado. La métrica básica es el módulo del gradiente de cada región ST (ya que este representa un medida de la actividad espacial).

**“Structural Similarity Index Measurement” (SSIM)** Esta medida se basa en una nueva filosofía introducida en [153, 154, 152]. En dichos trabajos, se argumenta que la principal función del sistema visual humano es extraer información estructural del campo visual, y el sistema visual humano está altamente adaptado a este propósito. Por lo tanto, proponen una medida de la distorsión estructural como una medida de la distorsión percibida. De acuerdo a [152], la información estructural es una característica que se representa a través de la estructura de los objetos, independientemente del nivel de luminancia y de contraste de la imagen.

### 12.2.3. Evaluación Objetiva - Métodos no-intrusivos

**E-Model** la relación entre diferentes características de las redes y de las secuencias multimedia y la calidad de la voz ha sido cuantificada en el E-Model [7], introducido por la ITU-T. Este modelo como se ha señalado en la literatura presenta un importante debilidad: asume que las características individuales como el retardo, el eco, la distorsión, etc. tienen efectos mutuamente independientes en la calidad percibida.

**PSQA** Este método utiliza un modelo basado en Random Neural Network (o RNN). Los resultados de tests subjetivos (DMOS) dependen básicamente de las características de la red (pérdidas, retardo, jitter) y de las características del multimedio (codec, tasa de bits, naturaleza del contenido). Si es posible establecer una relación entre estas características y la medida subjetiva DMOS, se puede aproximar el DMOS midiendo estos parámetros objetivos. Las RNN son un técnica de aprendizaje estadístico, que utiliza en una etapa de aprendizaje un conjunto de  $n$ -uplas (*características de la red y el multimedio, DMOS*) para construir este modelo. Posteriormente conociendo las características de la red y el multimedio se predice el DMOS.

## 12.3. Evaluación experimental y resultados

### 12.3.1. La maqueta de pruebas utilizada

La herramienta de software integra métodos de estimación objetivos intrusivos y no intrusivos. Los algoritmos evaluados son PESQ, EMBSD, MNB, PSQA en el caso de audio, y MSE, PSNR, SSIM, TSSDM PSQA para video.

Los detalles del software, su arquitectura y su implementación se encuentran en [31].

Para realizar los tests, se implementó una maqueta simple de pruebas que permite emular diferentes condiciones de una red de manera controlada. Esta maqueta se compone de dos equipos que funcionan como extremos de la red (servidor/cliente) conectados a través de un enrutador intermedio que permite simular pérdidas, retardo y jitter. La figura 12.3 presenta esta maqueta de pruebas.

Las pérdidas de paquetes en una red en general no son independientes, sino que ocurren en ráfagas debido a situaciones de congestión en la red. El modelo más simple para representar este comportamiento se encuentra desarrollado en [24], utilizando un modelo markoviano simple: el modelo de pérdidas de Gilbert. El modelo de pérdidas de Gilbert consiste en una cadena de Markov con dos estados, donde el estado 0 corresponde a un paquete recibido en el destino y el estado 1 a un paquete perdido. En la figura ??,  $p$  representa la probabilidad de pérdida de un paquete dado que el último paquete fue recibido correctamente, y  $q$  es la probabilidad de tener una correcta transmisión dado que el último paquete se perdió. Este modelo simple permite simular pérdidas en ráfagas.

El jitter y el retardo son controlados manipulando el tamaño del buffers y la capacidad de la interfaz ethernet del enrutador. Algunos paquetes auxiliares se insertan en el buffer de salida a los efectos de generar jitter, y el tamaño del buffer y la capacidad de salida se modifican para producir el retardo deseado.

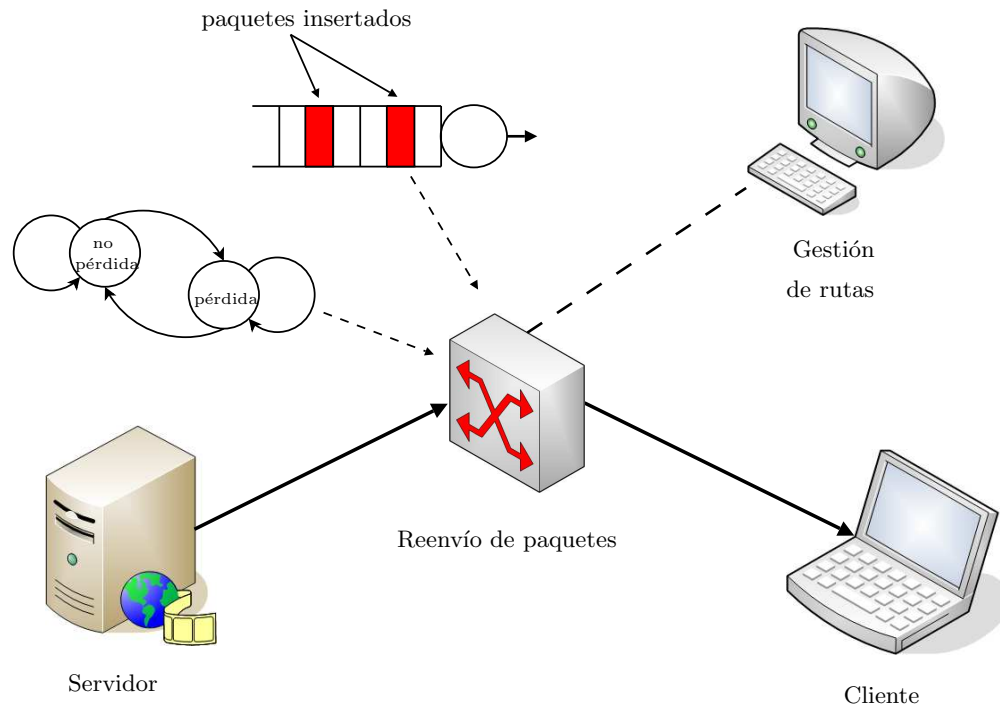


Figura 12.3: Maqueta de evaluación.

Las secuencias multimedia son 75 parejas original-distorsionada para el caso de video y 72 parejas de audio. Las secuencias de video de referencia fueron seleccionadas de acuerdo a las recomendaciones de ITU [66, 67] (40 secuencias cortas de 10-30 segundos) y clasificadas según el codec (MPEG1 y MPEG4) el nivel de movimiento (bajo, medio y alto). En audio, 24 secuencias cortas fueron grabadas y codificadas con diferentes codecs (PCM, GSM and G.723).

Las secuencias de referencia son enviadas a través de la maqueta de pruebas, configurando diferentes valores para los parámetros del enrutador a los efectos de cubrir un rango de características similares a los que se podría tener en Internet. Las secuencias obtenidas luego son utilizadas en los tests subjetivos (como se describió en la sección 12.2), obteniendo un conjunto final de datos:

$$\{sc_j, (p_0, f_1, \dots, f_i, \dots, f_n), DMOS\}, \quad (12.3)$$

donde  $sc_j$  es la  $j$ -ésima pareja de secuencias original-distorsionada,  $f_i$  es el valor de la  $i$ -ésima característica (por ejemplo tasa de pérdidas, largo medio de la ráfaga de pérdidas, jitter, codec, nivel de movimiento, etc..) y DMOS el resultado del test subjetivo correspondiente. Por último, parte del conjunto de datos fue usado para entrenar el algoritmo PSQA y para calibrar los test objetivos intrusivos, el resto de los datos fueron usados para validar.



### 12.3.2. Resultados de los test subjetivos

En la figura 12.4 se muestra la distribución de las muestras de datos utilizados (para audio y video) considerando: la tasa de pérdida, el largo medio de la ráfaga de pérdidas y el codec.

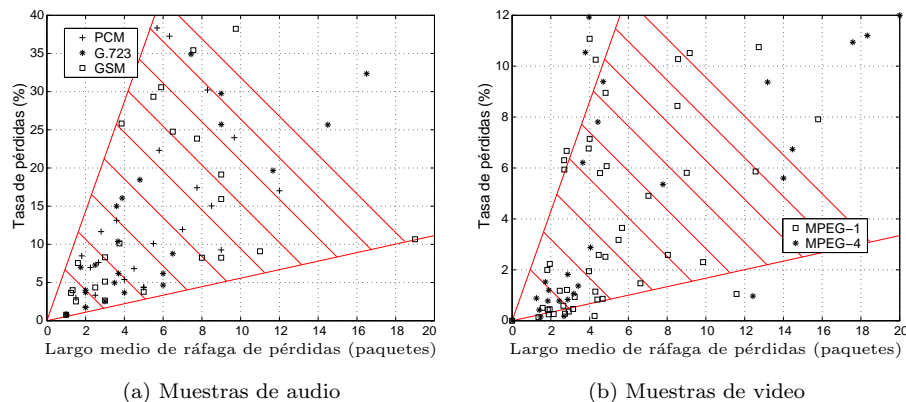


Figura 12.4: Distribución de los datos.

Para obtener una muestra de entrenamiento adecuada y calibrar adecuadamente los diferentes métodos, el *espacio de entradas* debe ser cubierto, particularmente para aquellos valores más usuales o donde es más difícil discriminar la calidad.

	DMOS medio	varianza media
<b>Audio</b>	3.04	0.36
<b>Video</b>	3.03	0.25

Cuadro 12.2: Características estadísticas de los tests subjetivos.

La tabla 12.2 presenta los resultados de los test DMOS subjetivos para audio y video, usando la escala de calidad 12.1(b). De acuerdo a las recomendaciones de la ITU para audio [65] y video [66], los tests subjetivos deben ser diseñados de manera tal que los resultados promedio deben estar en la mitad de la escala de calidad (a los efectos de evitar resultados sesgados). Estas recomendaciones también especifican el procedimiento para remover “outliers” de los resultados.

### 12.3.3. Evaluación de las diferentes técnicas

Para evaluar los algoritmos se utilizó el error medio absoluto (MAE), entre los valores estimados por los algoritmos y los valores reales de los tests subjetivos. Los test objetivos intrusivos no están en la misma escala que los valores DMOS (cada algoritmo usa su propia escala), por lo tanto antes de compararlos se calibran los test para llevarlos a la misma escala. Se utilizaron 70 % de las muestras para calibrar (entrenar en el caso de PSQA) y 30 % para validación. En audio, la relación usada es 80 % – 20 %.

## Audio

En la tabla 12.3(a) se presentan los valores de MAE para todos los algoritmos de audio de acuerdo a la escala de calidad DMOS 12.1(b).

Método	MAE
EMBSD	0.59
<b>PESQ</b>	<b>0.43</b>
PSQA	0.45
MNB	0.68

Método	CF
<b>PESQ</b>	<b>0.93</b>
PSQA	0.86

(a) Datos de entrenamiento      (b) Datos de validación

Cuadro 12.3: Error medio absoluto (MAE) factor de correlación (CF).

Una comparación gráfica de los algoritmos se muestra en la figura 12.5. Los resultados obtenidos para audio presentan el método PESQ como el más aproximado. Comparado con los otros métodos intrusivos, PESQ tiene una importante ventaja: incluye un algoritmos de re-sincronización temporal que permite una comparación más adecuada de las secuencias. Ante la presencia de pérdida de datos, una comparación directa de las secuencias puede resultar en un muy mal desempeño del algoritmo. PESQ es actualmente el algoritmo recomendado por la ITU para calidad percibida en voz [69]. El desempeño del algoritmo no intrusivo PSQA es muy cercana a la obtenida con PESQ, lo cual es muy interesante ya que muestra que los complejos modelos psico-acústicos propuestos por los diferentes algoritmos que se desarrollan en el *dominio perceptual* (PESQ, MNB, EMBSD) pueden ser aproximados por un mecanismo de aprendizaje estadístico como las Random Neural Networks.

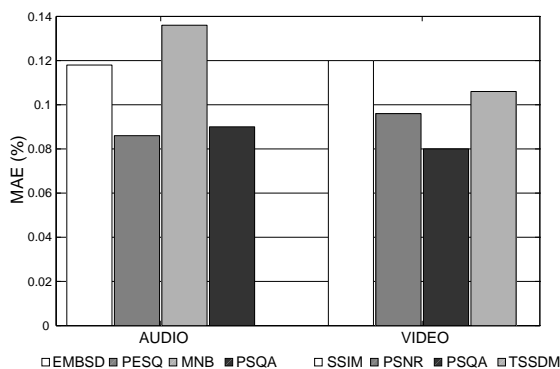


Figura 12.5: PQoS para audio y video, evaluación de los diferentes algoritmos.

La figura 12.6 muestra resultados obtenidos con PSQA (izquierda) y PESQ (derecha) con las muestras de datos de validación. Ambos algoritmos presentan una fuerte correlación con los resultados de los test subjetivos. Estos resultados muestran que el entrenamiento de las RNN obtuvo un modelo suficientemente aproximado como para reproducir una buena performance con los datos de

validación. La tabla 12.3(b) presenta factor de correlación (CF) entre el DMOS real y el estimado por PESQ y PSQA sobre los datos de validación (un valor cercano a 1 indica una alta correlación lineal).

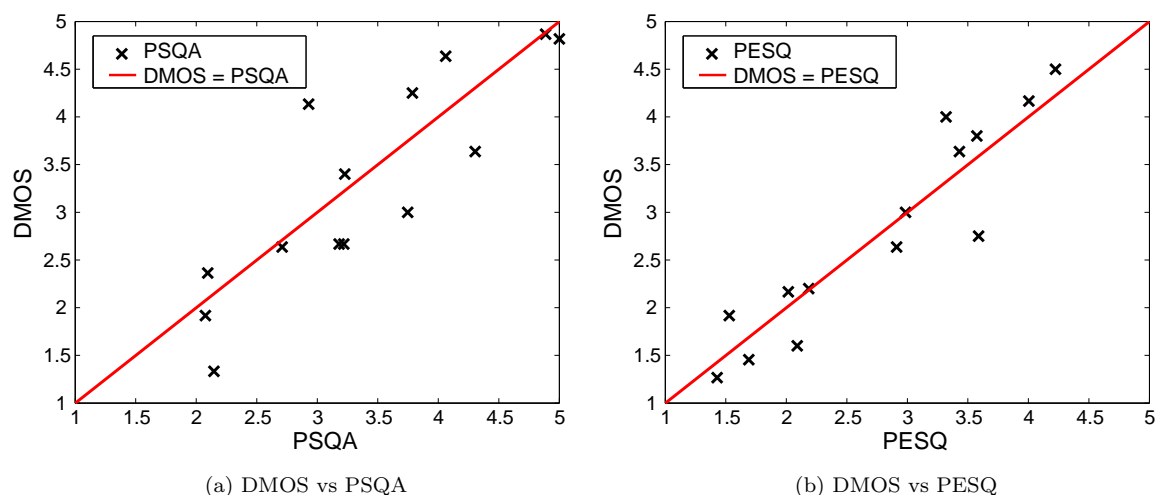


Figura 12.6: PQoS en audio, desempeño de PESQ y PSQA sobre la muestra de validación.

## Video

En el caso de video, PSQA es claramente el mejor método de los evaluados, no sólo porque brinda el menor error, sino también por el tiempo requerido para la estimación. La tabla 12.4 resume estas observaciones, presentando el error para la muestra de entrenamiento y el tiempo medio requerido para estimar la PQoS. La figura 12.7 muestra los diferentes algoritmos junto con la curva estimada que los relaciona (en el caso de PSQA es una recta indicando que Subjective DMOS = PSQA). La figura 12.7(c) confirma las observaciones previas respecto del desajuste de PSNR para evaluar la PQoS. Como se puede observar el mismo valor de PSNR corresponde a muchos valores diferentes de calidad percibida. En el caso de video no hay métodos estandarizados, lo que muestra que la PQoS para el video es todavía un problema abierto. Los métodos intrusivos presentados para video sufren del mismo problema de sincronización que se mencionó para audio.

Método	MAE	ACT (segundos)
SSIM	0.60	> 600
PSNR	0.48	≈ 20
<b>PSQA</b>	<b>0.40</b>	≈ <b>2</b>
TSSDM	0.53	> 1200

Cuadro 12.4: MAE y Tiempo medio de cálculo (ACT).

Para concluir con el análisis para el caso de video, se presenta en la figura 12.8 el resultado obtenido para PSQA sobre la muestra de validación. Como en el caso de audio, el modelo estimado con la RNN captura la compleja relación entre la calidad de servicio percibida y las características de la red y el multimedia.

### **Análisis de la PQoS a través de PSQA**

Una ventaja de los algoritmos basados en parámetros objetivos es que permiten analizar la influencia de las diferentes características sobre la PQoS. La figura 12.9 presenta la influencia los codec de voz (a) y el nivel de movimiento de video (b) sobre la calidad de servicio percibida como función de la tasa de pérdidas (utilizando PSQA como algoritmo de medida). Como era de esperar, las pérdidas en el caso de la codificación de la *G.711* (PCM puro, alta tasa de bits, sin un modelo predictivo que pueda ser afectado por las pérdidas) son “menos importantes”. En el caso de video, la evaluación confirma la idea de que el nivel de movimiento tiene influencia sobre la PQoS: las secuencias de video con tasas de movimiento altas presentan un decaimiento más rápido de la calidad de servicio percibida respecto de las pérdidas de paquetes que aquellos con niveles de movimiento bajo.

Por último, la figura 12.10 evidencia la influencia de la tasa de pérdidas y del largo medio de la ráfaga de pérdida (MLBL) sobre la calidad de servicio percibida de (a) voz y (b) video. La primera observación es que la calidad de servicio percibida del audio es menos sensible que la del video a las pérdidas de información. En segundo lugar, en ambos casos, la calidad de servicio percibida crece monótonamente con el largo medio de ráfaga (para una tasa de pérdidas fija), lo que implica que aparentemente desde el punto de vista perceptivo se prefiere que las pérdidas estén concentradas en pocos puntos a que estén repartidas sobre la secuencia.

## **12.4. Consideraciones sobre la PQoS y su integración con la estimación usando paquetes de prueba**

Un objetivo de la tesis es la estimación mediante técnicas de regresión de parámetros objetivos de la QoS de servicios de video por ejemplo, a partir de tráfico liviano de prueba. Lo que se busca en esta sección es realizar diferentes consideraciones y analizar los métodos de PQoS (principalmente el método PSQA que parece ser una metodología interesante por los resultados obtenidos). Además, se analizará como se pueden complementar las técnicas de PQoS con las herramientas de estimación de parámetros objetivos de QoS a través de regresión desarrollados en la tesis.

Los mayores problemas para aplicar los métodos de PQoS que se basan en test subjetivos es la dependencia del factor humano. Es decir, para que un test subjetivo sea realizado según los procedimientos estandarizados, es necesario mostrarle las secuencias de video distorsionadas a un grupo de personas y que estas evalúen la calidad percibida. Esto lleva a que métodos que tienen potencial para ser aplicados en línea como PSQA dependan de un entrenamiento fuera de línea, donde además es necesario generar de manera controlada las secuencias distorsionadas para un rango de valores (el rango que se presume que se puede observar en la red real) de las características de la red (pérdidas, retardo, jitter, etc.). Asociado a este problema aparece otro. No está claro en los modelos estudiados en PQoS cuánto impacta la correlación entre las características de la red. Por

ejemplo, asumamos que para generar los videos distorsionados se utiliza un modelo de Gilbert para generar pérdidas y ráfagas de pérdidas. En este modelo de la red se está asumiendo cierto tipo de correlación entre las pérdidas y las ráfagas de pérdidas. Eso influye en las muestras que se generan para entrenar el sistema. La pregunta es: si se realiza un modelo de un cierto camino en Internet y se generan pérdidas y retardos con algún modelo adecuado para ese camino con el cual se entrena por ejemplo una RNN: ¿ese entrenamiento es válido para otro camino en la red? La respuesta a esta pregunta no está clara en la literatura sobre PQoS. Si la respuesta a esta pregunta como nosotros pensamos es negativa, se deberá entrenar la RNN para cada camino de la red. Esto no sería un inconveniente si no interviniera el factor humano. Para cada camino se deben hacer sesiones de evaluación de grupos de personas con experimentos controlados. Evidentemente el costo de este proceso es muy elevado. En principio, surgen dos posibles soluciones a este problema. La primera es buscar entrenar en la propia red. Es decir se envía la secuencia de video por el camino entre por ejemplo un servidor de video y un usuario. Se mide entonces el desempeño que experimenta esa secuencia de video. Posteriormente se le pide al usuario que evalúe las diferentes secuencias de video obtenidas y las califique. Se debería repetir este experimento muchas veces en diferentes estados de la red para barrer los valores posibles de características que pueden aparecer en ese camino. El problema con este procedimiento es que se obtiene la PQoS adaptada a un usuario y no la opinión media de los usuarios del DMOS. Esto puede derivar en que si en un camino hay un usuario exigente y en otro uno poco exigente, las estimaciones de PQoS dirán diferentes valores de calidad para situaciones similares. En algunas aplicaciones, esto puede no ser relevante y se podría usar este mecanismo. La segunda alternativa y que parece más prometedora, es entrenar la RNN utilizando en lugar de la opinión de usuarios obtenida de un test subjetivo, utilizar la evaluación de PQoS que brinda un test tipo “null reference”. Es decir, para entrenar en un camino la RNN se podría enviar la secuencia de video, medir sobre ella sus parámetros de desempeño (retardo, pérdidas, etc.) y a la secuencia recibida asignarle un valor de PQoS que brinda un test sin referencia por ejemplo de los que tienen un modelo en el campo perceptivo. De esta forma se elimina el factor humano del problema. Esta un área interesante para explorar en el futuro y evaluar los resultados que se obtienen.

Otra debilidad de las técnicas de estimación de PQoS es que es necesario enviar el video para evaluar la PQoS, por ejemplo en PSQA.

En los capítulos anteriores se vio que la estimación de los parámetros objetivos de la QoS de una aplicación puede hacerse a partir los tiempos entre arribos de los paquetes de prueba.

Los conceptos y los diferentes algoritmos de PQoS analizados en las secciones anteriores pueden complementarse con los métodos de regresión vistos en la tesis. Existen varias formas de integrarlos.

Una primera forma de integración puede ser la siguiente. Con la estimación de las variables  $Y$  de los modelos de regresión vistos puede ser entradas por ejemplo para una RNN entrenada para estimar PQoS. Eso evitaría tener que enviar el video todo el tiempo para monitorear la calidad de servicio percibida.

Otra forma de integración es la siguiente. Una vez entrenada una RNN, esta permite clasificar regiones de los parámetros objetivos de la QoS (pérdidas retados, etc.) y del video. Por ejemplo podría decirse que para obtener un PQoS con un DMOS entre 4 y 5 implica pérdidas en cierto rango, retardo en cierto rango para un video MPEG4 y de alto nivel de movimiento. Una vez definidas estas zonas de los parámetros objetivos, se pueden utilizar las técnicas vistas de envío de paquetes

de prueba livianos para ver en qué región de PQoS se estaría. Para implementar este algoritmo en lugar de hacer una regresión de la variable  $Y$  como hasta ahora, se pueden usar técnicas de clasificación. En la siguiente sección se explica este procedimiento como ejemplo de esta posible forma de integración.

## 12.5. Clasificación y selección de características en regiones de QoS

### 12.5.1. Clasificación en zonas de QoS

En muchos casos no es necesario aproximar precisamente el valor de un parámetro de calidad de servicio, pero si estimar regiones donde hay ciertos comportamientos de un conjunto de parámetros de QoS. Por ejemplo, para ciertos servicios mediante técnicas de PQoS se puede haber determinado que, dependiendo en los rangos que caen el retardo medio, y los valores de pico del retardo se tendrán niveles diferentes de PQoS. Se podría hacer una regresión de estos dos valores, ver en qué zona caen y obtener de allí cual sería la PQoS. Sin embargo, requiere un costo computacional menor saber clasificar de acuerdo al estado de la red en qué región estarán estos dos parámetros sin conocer sus valores exactos.

Para evaluar este procedimiento, se propone usar SVM para clasificar en zonas de QoS. Si bien en los casos que se analizan a continuación con datos simulados se trabaja con dos parámetros de QoS, se pueden agregar a estos otros parámetros como las pérdidas de paquetes o el jitter, etc..

En la fase de entrenamiento se definen zonas sobre el conjunto de variables  $Y_i$  de interés y con esta información se entrena un clasificador SVM. La  $X$  será un estimador del estado de la red extraído de los paquetes de prueba. Posteriormente usando el modelo aprendido con SVM, cuando se hace una nueva medida del estado de la red  $X$  se clasifica y se estima en qué zona de QoS se encuentra. Con esto se define a cuál subconjunto de variables  $Y_i$  pertenece. Si esta clasificación de las  $Y_i$  proviene de la PQoS asociada, se puede predecir entonces a partir de los paquetes de prueba la PQoS.

Se utilizará para evaluar este procedimiento el mismo conjunto de datos simulados con ns-2 en los capítulos anteriores pero ahora se medirá el retardo medio ( $Y_1$ ) y los valores de pico del retardo medido como el percentil 90 % sobre la distribución del retardo de los paquetes de video ( $Y_2$ ). Estas dos variables para el ejemplo se clasifican arbitrariamente en los siguientes grupos:

Clase	$Y_1$	$Y_2$
Clase 0	$< 0.05 \text{ s}$	$< 0.2 \text{ s}$
Clase 1	$0.05 \text{ s} < Y_1 < 0.1 \text{ s}$	$0.2 \text{ s} < Y_2 < 0.3 \text{ s}$
Clase 2	$0.1 \text{ s} < Y_1 < 0.15 \text{ s}$	$0.3 \text{ s} < Y_2 < 0.5 \text{ s}$
Clase 3	$0.15 \text{ s} < Y_1 < 0.2 \text{ s}$	$0.5 \text{ s} < Y_2 < 0.7 \text{ s}$
Clase 4	$0.2 \text{ s} < Y_1$	$0.7 \text{ s} < Y_2$

A los efectos de clasificar, se consideró como vector  $X$ , un vector con las siguientes 13 componentes de la distribución de la cola: media de la cola, varianza, percentil 10 %,20 %,...,90 %,95 % y la probabilidad de que la cola esté vacía.

Se utilizó SVM para clasificar con las mismas muestras de entrenamiento y test utilizadas antes. El resultado que se obtuvo, se muestra en la figura 12.11. En los 120 puntos de test se clasificó bien en casi el 90 % de los puntos (107 puntos bien clasificados de 120). Se debe observar también que los errores de clasificación se dieron en puntos ubicados exactamente sobre la frontera de las zonas. Por ejemplo en los primeros puntos mal clasificados, los puntos 10 y 11 en el gráfico, el valor real del retardo medio y el percentil 90 % del video para ellos es:

10- retardo medio: 0.202 s, percentil 90 % 0.67 s

11- retardo medio: 0.198 s, percentil 90 % 0.708 s

Se debe observar que la clase 4 a la que pertenecen está limitada por retardo medio mayor a 0.2 s y percentil 90 % del retardo mayor a 0.7 s. Es decir que el punto 10 pertenece a la clase 4 porque superó en 0.002 s el retardo medio límite y el punto 11 porque el percentil 90 % del retardo superó en 0.008 s el valor frontera. Estas diferencias están dentro del error de medición por lo que el punto podría ser clasificado en cualquiera de las clases.

### 12.5.2. Selección de variables utilizando F-Score

La pregunta que se analiza en esta sección, es con cuántas características de las 13 de la distribución de la cola consideradas antes se puede tener una buena clasificación. Dicho de otra forma, son necesarias todas las variables o se puede prescindir de algunas de ellas y no afectar el desempeño de la clasificación.

La selección de características es un problema importante por diferentes razones. Por un lado, reducir el número de características disminuye el tiempo de aprendizaje y los requerimientos de almacenamiento. Además, eliminar características que son irrelevantes, y mantener aquellas que son más significativas aumenta la precisión y el desempeño de los algoritmos. Por último, encontrar cuáles son las características más relevantes puede contribuir a comprender mejor el problema y descubrir la información mas relevante que se maneja en los datos.

Existen muchas técnicas para seleccionar características en un problema de aprendizaje. Algunos métodos están fuertemente determinados por el clasificador que se vaya a usar, mientras que otros son independientes del clasificador y por tal motivo se los conoce como métodos de filtrado.

En este trabajo se utilizará un método de filtrado. Existen diferentes métodos de filtrado basados en propiedades estadísticas de los datos. Por ejemplo, hay métodos basados en calcular coeficientes de correlación o basados en el criterio de Fisher [95]. Estos criterios tienen como desventaja que no aprovechan las potencialidades del clasificador a utilizar. Sin embargo, tienen como importante ventaja que son simples, efectivos y fáciles de implementar lo cual es una característica importante para aplicaciones en línea.

En este trabajo se utilizará en particular el criterio de Fisher (también conocido como F-score). El criterio de Fisher para una característica se define de la siguiente forma. Dado un conjunto  $X$  con dos clases, se denominará a las instancia de la clase 1 por  $X^1$  y a las de la clase 2 por  $X^2$ . Se llamará  $\bar{x}_j^k$  al promedio de la  $j$ -ésima característica en  $X^k$ . El F-score de la  $j$ -ésima característica es:

$$F(j) = \frac{(\bar{x}_j^1 - \bar{x}_j^2)^2}{(s_j^1)^2 + (s_j^2)^2} \quad (12.4)$$

$$s_j^k = \sum_{\mathbf{x} \in X_k} (x_j - \bar{x}_j^k)^2 \quad (12.5)$$

El numerador de esta expresión mide como se separan en media las clases con esta características y el denominador es una medida de la variabilidad de esta característica en cada clase.

Para el caso de tener una clasificación en más de dos clases se extiende el F-score definido antes de la siguiente forma. Dado un conjunto  $X$  con  $m$  clases, se denominará a las instancia de la clase  $k$  por  $X^k$  y al  $|X^k| = l_k$ ,  $k = 1, \dots, m$ . Se notará  $\bar{x}_j^k$ ,  $\bar{x}_j$  al promedio de la  $j$ -ésima característica en  $X^k$  y  $X$  respectivamente. El F-score de la  $j$ -ésima característica es:

$$F(j) = \frac{\sum_{k=1}^m l_k (\bar{x}_j^k - \bar{x}_j)^2}{\sum_{k=1}^m s_j^k} \quad (12.6)$$

$$s_j^k = \sum_{\mathbf{x} \in X_k} (x_j - \bar{x}_j^k)^2 \quad (12.7)$$

Se analizará entonces, para cada característica el resultado con F-score y luego se verá para este caso que relación hay entre F-score y el resultado obtenido clasificando con svm usando cada característica por separado.

Los resultados de F-score para cada característica de la distribución de la cola, se resumen en la siguiente tabla:

Característica	F-score
media	0.67
varianza	0.68
percentil 10 %	0
percentil 20 %	0
percentil 30 %	0.001
percentil 40 %	0.0037
percentil 50 %	0.65
percentil 60 %	0.71
percentil 70 %	0.66
percentil 80 %	0.66
percentil 90 %	0.66
percentil 95 %	0.66
prob. de cola vacía	0.56



Como se puede ver los valores de F-score generan dos grupos entre las características las de valores muy bajos y las de valores entorno al 0,6. Esto lleva a pensar en que se podría filtrar las variables con valores menores que 0.5. Sobre los restantes valores el método no dice nada muy concluyente. Para verificar esto y ver si existe alguna relación entre los valores que da F-score y la clasificación que se logra con SVM, se analiza a continuación para cada característica independiente la clasificación usando SVM. La tabla siguiente resume el porcentaje de aciertos usando cada característica individualmente para clasificar:

Característica	% aciertos (puntos correctos/puntos totales)
media	89.1 % (107/120)
varianza	83.3 % (100/120)
percentil 10 %	48 % (58/120)
percentil 20 %	48 % (58/120)
percentil 30 %	48 % (58/120)
percentil 40 %	48 % (58/120)
percentil 50 %	76 % (91/120)
percentil 60 %	89.1 % (107/120)
percentil 70 %	89.1 % (107/120)
percentil 80 %	87.5 % (105/120)
percentil 90 %	88.3 % (106/120)
percentil 95 %	84.2 % (101/120)
prob. de cola vacía	75 % (90/120)

Es interesante observar que el valor medio de la cola o algunos percentiles por separado permiten clasificar con la misma precisión que si se usan todas las características juntas. El caso de utilizar sólo el valor medio de la cola se muestra en la figura 12.12.

De las características que tenían un valor de F-score en el grupo “alto” las tres que tuvieron peor performance fueron la probabilidad de cola vacía, el percentil 50 % y la varianza.

En la figura 12.13 se muestra el caso de clasificar utilizando sólo la probabilidad de cola vacía. Como se marca en la figura, los errores adicionales en este caso se dan en la mala clasificación en las clases 2, 3 y 4 ya que en varios casos las confunde. Esto ya había sido observado cualitativamente en las figuras 9.7 y 9.8. En ellas se observa que la probabilidad de cola vacía separa bien los valores de los bajos del retardo medio del video (una de las variables de clasificación), pero los valores intermedios y altos de retardo se encuentran relativamente más juntos y por lo tanto es posible cometer más errores en este caso.

En cambio en la figura 12.14 se ven los errores adicionales que se cometen en el caso de utilizar el percentil 50 %. En este caso los errores de clasificación se dan entre las clases “bajas”. Esto también se puede analizar cualitativamente a partir de las funciones distribución empírica del retardo medio

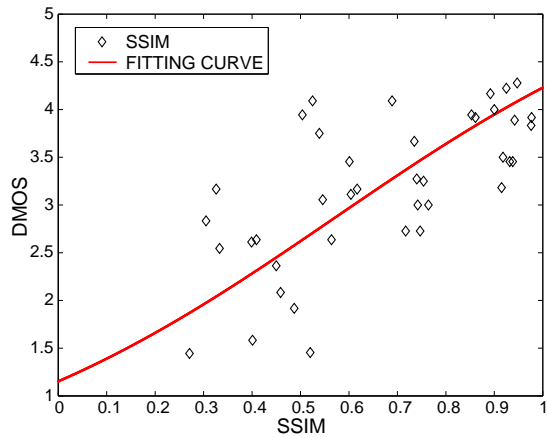
del video (figuras 9.7 y 9.8). En ella se ve que el valor de este percentil para las clases medias y bajas es 0 o próximo a cero y por lo tanto estos valores pueden ser mal clasificados. También se ve que para percentiles menores a 50 el error crece rápidamente ya que por debajo de 40 % en los datos mostrados en la figura el percentil vale 0 para todos los casos.

Por último en la figura 12.15 se analiza el caso de clasificar usando sólo la varianza de la cola. En este caso se clasifican mal 6 puntos adicionales que corresponden a valores intermedios clasificados en la clase 2 colocados en la clase 1.

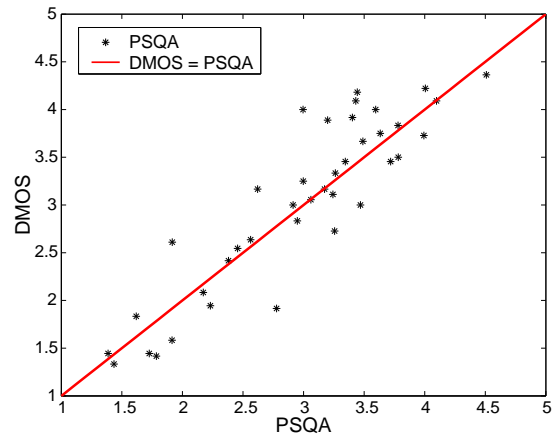
Si bien es cierto que parte de la información aportada por el análisis anterior podría haber sido extraída del análisis cualitativo de los datos, F-score permite automatizar al menos un primer filtro para eliminar características que aportan poca información. No debe olvidarse que el análisis de este capítulo fue realizado con un conjunto de datos obtenidos mediante simulación en un simulador de redes. Cuando se pase al caso de la red real no necesariamente se mantendrán para los diferentes caminos y casos las mismas características que permitan hacer regresión o clasificación. Sin embargo, los procedimientos desarrollados aquí por un lado son muy auspiciosos en cuanto a los resultados obtenidos y por otro permitirán analizar los casos y decidir qué características utilizar de acuerdo a aquellos parámetros que mejor se adapten a cada caso.

## 12.6. Conclusiones

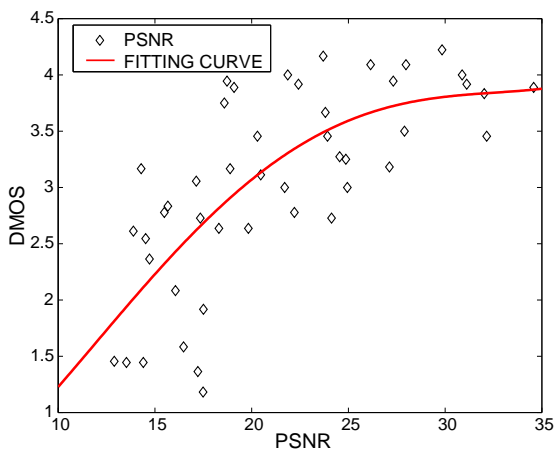
En este capítulo se analizaron diferentes metodologías de estimación de la calidad percibida por un usuario. Se compararon diferentes métodos de PQoS y se vio con cuáles se obtiene mejor desempeño. También se realizó una evaluación crítica de estas técnicas viendo sus desventajas para implementarlas en una aplicación en línea. Por último se analizó como integrar las técnicas de estimación de la calidad de servicio utilizando paquetes de prueba propuesta en la tesis con algunos métodos de PQoS. Sin duda, lo analizado en este capítulo muestra principalmente líneas para trabajos futuros que integren las técnicas de estimación presentadas en la tesis con las técnicas desarrolladas para analizar la PQoS.



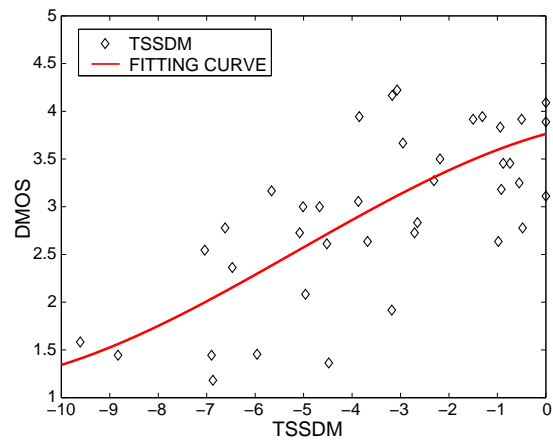
(a) DMOS vs SSIM



(b) DMOS vs PSQA



(c) DMOS vs PSNR



(d) DMOS vs TSSDM

Figura 12.7: PQoS en video, desempeño de los diferentes algoritmos.

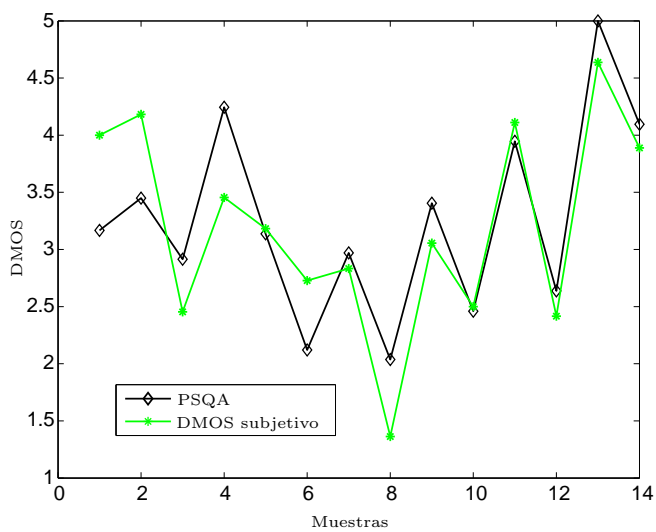


Figura 12.8: DMOS y PSQA - muestra de validación.

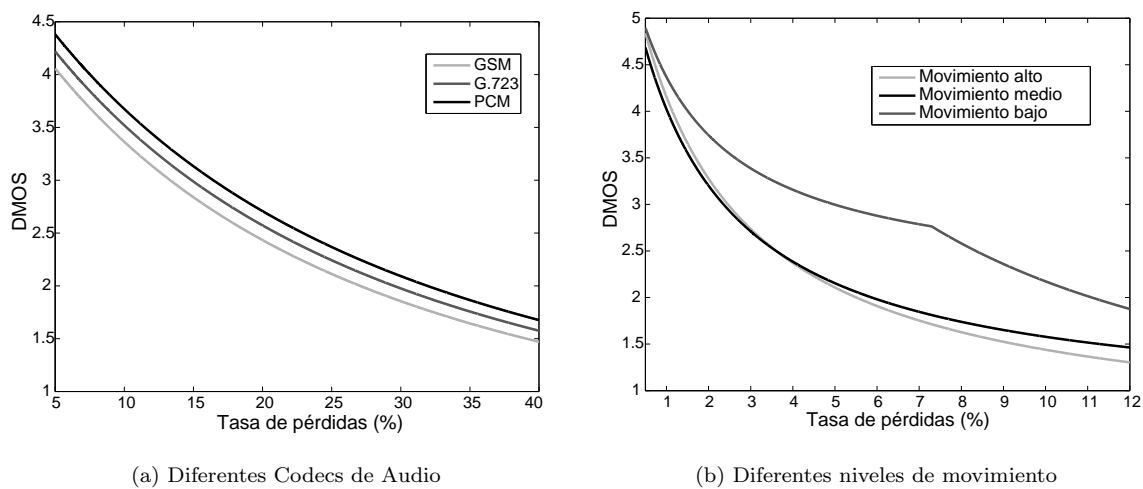
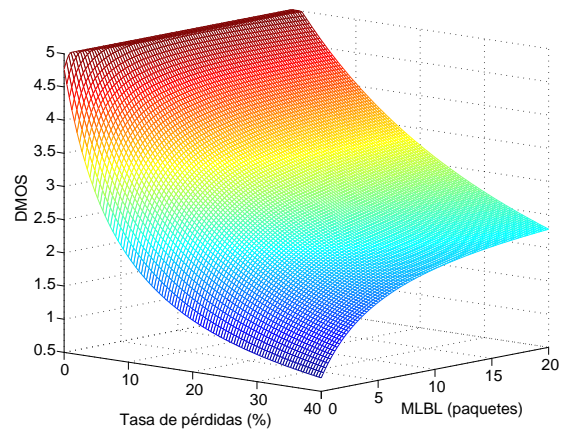
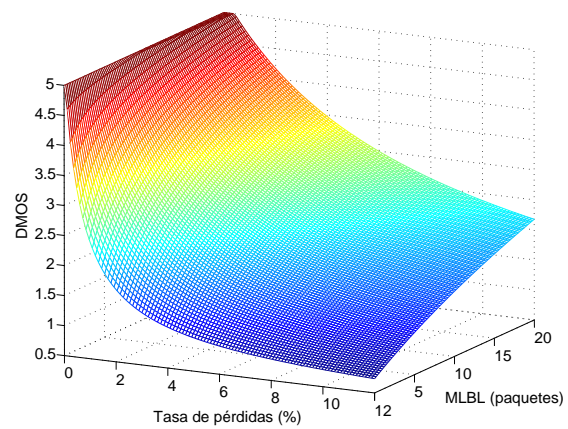


Figura 12.9: DMOS vs tasa de pérdida (MLBL = 5 paquetes).



(a) Evaluación de Audio (codec PCM)



(b) Evaluación de Video (codec MPEG4)

Figura 12.10: DMOS vs tasa de pérdidas y largo medio de ráfaga de pérdidas.

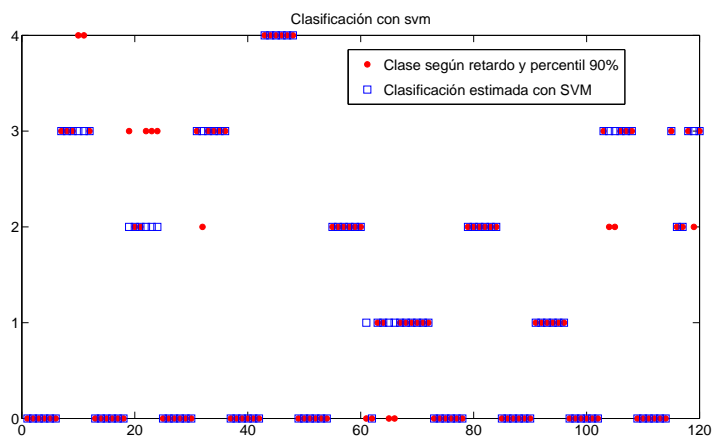


Figura 12.11: Clasificación utilizando 13 características de la distribución de la cola

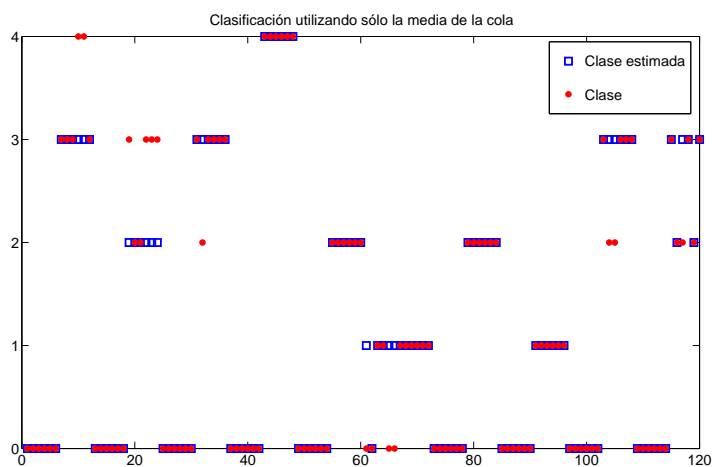


Figura 12.12: Clasificación utilizando sólo el valor medio de la distribución de la cola

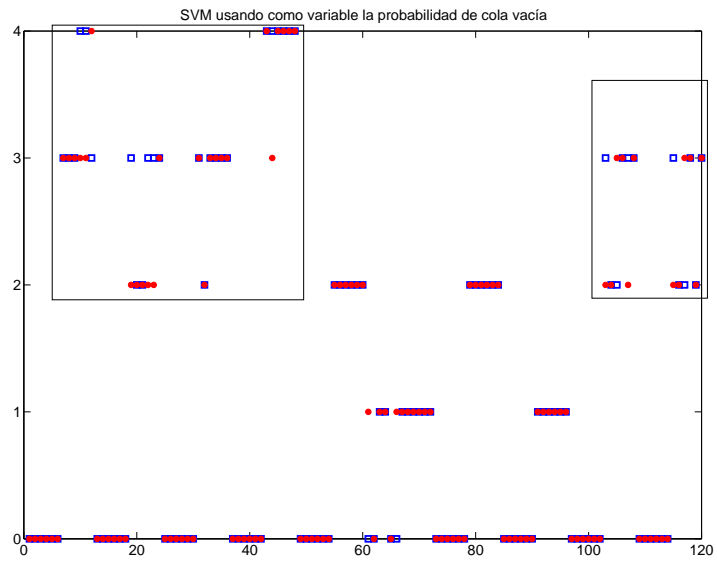


Figura 12.13: Clasificación utilizando sólo la probabilidad de cola vacía

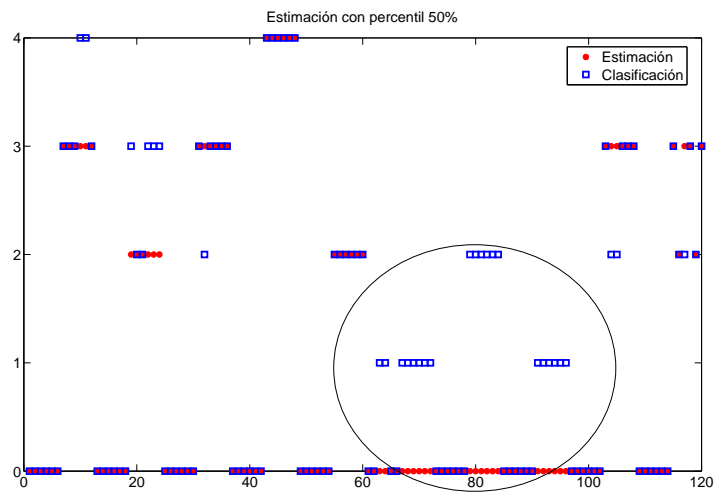


Figura 12.14: Clasificación utilizando sólo el percentil 50 % de la distribución de la cola

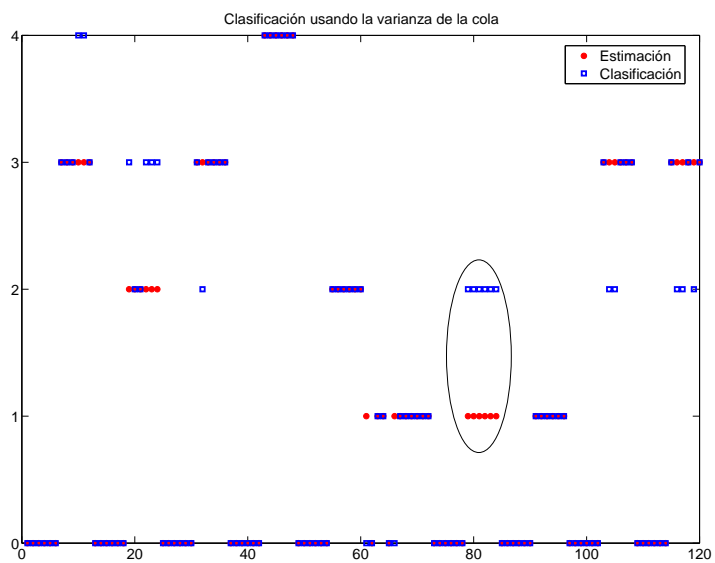


Figura 12.15: Clasificación utilizando sólo la varianza de la distribución de la cola



## Parte III

# Tarificación en redes con calidad de servicio

## Capítulo 13

# Tarificación en redes con calidad de servicio asegurada

### 13.1. Introducción

Esta parte de la tesis analizará el problema de brindar servicios de calidad asegurada en una red pero proponiendo, además, un mecanismo de tarificación que permita al operador justificar la inversión en este tipo de redes y optimizar su rentabilidad. Los resultados principales de esta parte de la tesis han sido publicados en [14, 15].

Como se dijo en la introducción de la tesis además de dificultades técnicas existen dificultades en cuanto a modelos de negocio de los operadores para desplegar ciertos servicios. El sesgo hacia la tarifa plana, impuesto primero en el acceso a Internet y luego extendido a otros servicios, genera dos problemas para desplegar redes con calidad de servicio asegurada. Por un lado, el operador pierde incentivos para la inversión de redes con mejor calidad de servicio y por otro, el usuario tampoco tiene incentivos para realizar un uso adecuado de los recursos compartidos en las redes.

En esta parte de la tesis se propone un mecanismo de subastas en línea para acceder a ciertos servicios premium que un operador puede ofrecer sobre el acceso a Internet. Los usuarios requieren para acceder a un servicio del uso de recursos de la red (normalmente cierto ancho de banda en el camino entre el usuario y el servidor). La idea será entonces que mediante una subasta los usuarios que estén dispuestos a pagar más por estos recursos sean los que accedan.

La subasta de capacidad en tiempo real ha sido considerada por diversos autores [42, 49, 87, 97, 135, 127], y para diversas aplicaciones: diffserv, control de admisión, acceso celular a redes 2G y 3G, VPNs, etc. Una parte importante de estos trabajos se han concentrado en la aplicación de herramientas de la teoría de juegos, en particular para brindar incentivos a los oferentes para que revelen su verdadera utilidad. La teoría de subastas [82] analiza y propone este tipo de mecanismos para subastar un único recurso, pero es un problema sensiblemente más complejo extender estos mecanismos a la topología de una red.

Dramitinos et al. [49] proponen realizar subastas para acceder a redes 2.5/3G. Se plantea utilizar mini-subastas de tipo GVA (Generalized Vickrey Auctions [82]) slot a slot para asignar recursos en un enlace 2.5/3G. Como el usuario no puede estar subastando slot a slot, el sistema plantea

un conjunto de funciones utilidad “típicas”. El usuario al elegir el servicio a usar (acceso a datos elásticos, video streaming, etc.), elige implícitamente la función utilidad y suministra el precio de su oferta que es un parámetro de la función utilidad. Con estas funciones utilidad el sistema asigna los slots a los usuarios, rematando entre todos los usuarios que están accediendo o quieren acceder al enlace en cada slot.

Reichl et al. [127], analizan el problema de asignación de recursos para conexiones largas que se desarrollan en múltiples períodos y proponen en [126] un mecanismo por el cual en cada subasta periódica compiten todos los usuarios (tanto los que ya tienen el recurso como los nuevos oferentes) pero si quien tiene el recurso asignado lo pierde, tiene la posibilidad de una segunda oferta para procurar conservarlo.

En las dos propuestas mencionadas antes aparece el problema de las consideraciones intertemporales de este tipo de subastas. Es decir, una vez que asigno un recurso a un usuario ¿el usuario lo preserva hasta que finalice su conexión? ¿o tiene que volver a ofertar periódicamente y competir por el recurso que ya tiene asignado? La solución adoptada por los trabajos previos ha sido de una forma u otra hacer competir periódicamente a todos los usuarios (incluido el usuario que lo tiene asignado en ese momento) por el recurso. Esto se hace por dos razones. Por un lado si aparece una oferta mejor y el recurso se encuentra ocupado, esta oferta no se pierde. Pero sobre todo esto se hace porque simplifica el análisis del problema. No es necesario mirar hacia el futuro, la decisión es siempre con las ofertas actuales.

La simplificación planteada en los trabajos anteriores respecto al problema intertemporal, resulta poco práctico y poco adecuado en el contexto de esta tesis. Por ejemplo para un usuario que quiere ver una película, no parece razonable que la misma pueda ser interrumpida, aunque eventualmente se le deje volver a subastar para poder finalizar de verla. Aunque se usen funciones de utilidad que penalicen la pérdida de períodos de conectividad, la calidad y la no interrupción del servicio no están garantizadas en los trabajos referidos. Garantizar el servicio con la calidad requerida es una condición funcional básica en el contexto de la tesis.

Ahora bien, si el operador le asigna un recurso y le asegura su uso a un usuario hasta que finalice el servicio, significa que el operador debe asumir el riesgo de las futuras subastas y la potencial pérdida de ganancia si aparecen ofertas mejores. Optimizar la ganancia del operador teniendo en cuenta este riesgo transforma el problema de subastas en un problema de optimización dinámica estocástica que en la tesis se formulará como un “Markov decision process” (MDP) [5, 123]. Sobre este tipo de problemas se trabajará en el caso de la subasta de los recursos de un enlace y se propondrá una forma de resolverlo.

Otro aspecto importante que no ha sido resuelto totalmente en la literatura es la subastas de los recursos de una red. Las propuestas existentes en este marco requieren que el usuario (o un agente intermediario actuando en su nombre), haga ofertas separadas por cada uno de los recursos internos de la red. En particular, el mecanismo denominado “Progressive Second Price” (PSP) [87] requiere que cada jugador coordine sus ofertas en diferentes nodos de la o las rutas que está subastando, y de esta forma cada nodo realiza una subasta utilizando los mecanismos de asignación y de cobro de un único recurso. PSP tiene además un tiempo de convergencia largo, y por tal motivo ha sido mejorado en la propuesta del mecanismo denominado “*multibid*” por Maille et al. [97]. Sin embargo, este último mecanismo sólo puede extenderse a redes con topología de árbol. Otra propuesta orientada a subastar árboles múlticast o VPNs es desarrollada por Courcoubetis et

al. [42]. Este mecanismo está basado en subastas tipo holandés (Dutch auctions). Este mecanismo asume que los usuarios interesados en un camino deberían tratar de reservar capacidad haciendo ofertas simultáneamente por todos los enlaces que lo constituyen.

El objetivo en esta tesis es que el usuario no deba conocer la topología de la red, ni realizar ofertas coordinadas por diferentes partes de la misma. Se busca que los recursos de la red sean asignados a las ofertas de los usuarios, los cuales hacen una oferta global por el servicio que desean. Además se desea por razones de escalabilidad un mecanismo de subasta que pueda ser implementado de manera distribuida en la red sin que un agente central sea quien tome las decisiones. Por estas razones las propuestas anteriores no son aplicables en este contexto. En el capítulo siguiente se propondrá un mecanismo de subastas para una red que cumple con los objetivos antedichos, en este se comenzará estudiando el caso de un enlace.

La primera decisión a tomar para analizar la aplicación del mecanismo de subastas, es sobre qué arquitectura de red de servicios con calidad garantizada trabajar como caso de análisis. Luego de estudiar diferentes arquitecturas, se decidió trabajar sobre dos propuestas, que se tomarán como ejemplo para pensar en este problema. Estas dos propuestas de arquitectura son la arquitectura overlay denominada: Service Overlay Network (SON) y las redes triple o cuádruple play.

La arquitectura SON presenta dos aspectos que la hacen atractiva para tomarla como ejemplo. Por un lado es una arquitectura multidominio, lo que permitiría la generación de operadores que ofrecieran servicios premium a través de más de un dominio de Internet. Por otra parte, es una arquitectura overlay. Las redes overlay están teniendo un crecimiento muy importante en Internet y han generado varias de las propuestas más innovadoras en estos últimos años como por ejemplo: las redes de servidores de contenido distribuidos, las redes “peer to peer”, redes comerciales como Akamai, varias redes de telefonía sobre IP, redes de distribución de tv por Internet, etc.. Por lo tanto, resulta atractivo, pensar en una arquitectura overlay multidominio como posible ejemplo de aplicación de los mecanismos de subasta propuestos.

La arquitectura triple play tiene como gran atractivo que es la arquitectura comercial sobre la que varios operadores están desplegando servicios de telefonía, video, datos, etc.. Esta arquitectura en realidad se puede pensar también como una red overlay. En este caso los nuevos servicios se superponen a la infraestructura básica de acceso a datos y se les reserva a ellos ciertos recursos en el acceso y cierto ancho de banda en caminos de la red. Es por tanto también un ejemplo atractivo sobre el que pensar mecanismos de tarificación mediante subasta para acceso a servicios premium.

Estos dos ejemplos, como se verá más adelante, tienen además una particularidad que los diferencia y que también los hace casos interesantes para este análisis. En las redes SON, el recurso de más alto valor sobre el que interesa optimizar la ganancia es el “backbone”. Esto se debe a que el operador SON invierte contratando ancho de banda para este backbone a diferentes ISPs de forma de tener caminos de ancho de banda asegurado a través de ellos. En el caso de las redes triple play en cambio, en general el backbone está sobredimensionado y el recurso que hay que optimizar es el acceso. Por lo tanto, esta dualidad permite mirar diferentes aspectos del problema y buscar una solución general.

En la sección 13.2 se explican las características de las redes overlay y en particular de la arquitectura SON. En la sección 13.3 se explican las ideas principales de las redes triple play. En la sección 13.4, se hace una breve introducción a los mecanismos de subastas. En la sección 13.5 se explica el problema a estudiar y la notación que se utilizará. Posteriormente, se analiza y se ven

ejemplos de como abordar el problema de rematar recursos en un sólo enlace. En la sección 13.6 se presenta una aproximación que permite resolver el problema de rematar recursos en un enlace. Por último en la sección 13.7 se plantean algunas conclusiones de este capítulo.

Como se mencionó, se pospone para el siguiente capítulo el análisis de la subasta de los recursos de una la red.

## 13.2. Redes overlay

El concepto de red overlay ha ido evolucionando a lo largo del tiempo. Internet en sus orígenes fue una red overlay sobre la red telefónica pública. Posteriormente, el transporte IP se ha ido transformando en la infraestructura sobre la que se transporta telefonía, datos, video, etc.. Internet ha desarrollado sus propias redes overlay con diferentes propósitos en algunos casos comerciales y en otros experimentales: redes de distribución de contenido (CDN), redes peer to peer, overlay multicast, overlay routing, etc..

Clark et al.[36] definen una red overlay como un conjunto de servidores desplegados a través de internet que:

- Proveen infraestructura para una o más aplicaciones
- Toman la responsabilidad de manejar y enviar los datos de la aplicación en forma diferente o en competencia con lo que hace la infraestructura básica de Internet
- Puede ser organizada y operada de forma coherente por terceras partes (incluyendo comunidades de usuarios finales)

Desde el punto de vista del modelo de capas OSI, si bien hay matices entre diferentes propuestas de redes overlay, se puede decir que operan entre la capa de transporte y la capa de aplicación. En este tipo de redes, se tienen servidores que toman funciones que antes estaban en los enrutadores.

Un interesante análisis de las redes overlay y su impacto en el futuro de Internet se puede ver el trabajo de Clark et al. [36]. Algunos trabajos, plantean una arquitectura overlay para solucionar problemas del enrutamiento en IP. Un ejemplo es RON (Resilient Overlay Networks) [6]. Otro trabajo de interés es QRON (QoS aware routing in overlay networks) [89] que aborda el problema de ofrecer ruteo con QoS sobre una red overlay. OverQoS [137], es una arquitectura propuesta para proveer a Internet de QoS usando redes overlay. En otros trabajos se busca utilizar una arquitectura overlay para optimizar la performance de cierto tipo de aplicaciones específicas. Este es el caso de Overphone [125]. En el trabajo de Wijnants et al. [156], los autores analizan una arquitectura overlay flexible orientada a procesar flujos multimedia por ejemplo haciendo transcoding o mixing de señales, etc.. Existen también, varios trabajos que enfocan el problema de overlay multicast. Overcast [75] es un ejemplo. En el trabajo de Parmer et al. [119], se comparan varios algoritmos de multicast en redes overlay en función de la latencia, complejidad y carga de enlaces físicos. Un análisis de algoritmos overlay multicast también es abordado en el trabajo de Fahmy et al. [53]. En el trabajo de Zhu et al. [164], se desarrollan algoritmos multicast usando network coding en una arquitectura overlay.

Resumiendo, las redes overlay están en plena expansión y parecen señalar un nuevo cambio en el futuro de la arquitectura de Internet. Han generado varias de las más importantes innovaciones de los últimos años en Internet y presentan oportunidades de nuevas formas de innovar desde los extremos en Internet. Han generado también algunos nuevos modelos de negocio, y cuestionan aún más los viejos modelos de negocio de los operadores de Internet.

### 13.2.1. Service overlay networks

Como vimos en la sección anterior, existen diferentes propuestas sobre redes overlay. Este trabajo utilizará como un caso de análisis, la arquitectura conocida como Service Overlay Network (SON). Esta arquitectura fue planteada en el trabajo de Duan et al. [51].

En dicho trabajo, se analiza cómo establecer una arquitectura que brinde servicios con requerimientos de QoS multidominio, y evite los problemas potenciales de performance en las conexiones entre diferentes dominios autónomos.

La arquitectura como se muestra en la figura 13.1 se basa en “service gateways”. Estos realizan funciones de reenvío de datos y funciones de control de la red. Las conexiones entre los service gateways son suministradas por los ISPs correspondientes mediante un Service Level Agreement de tipo “pipe SLA model” (brinda una conexión con ancho de banda garantizado a través de un ISP).

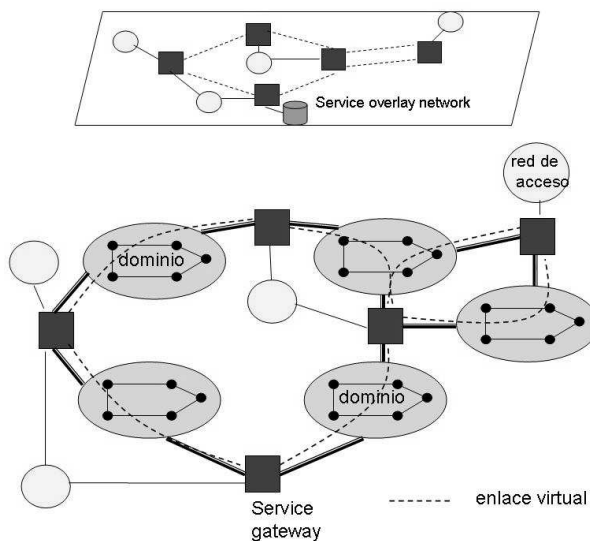


Figura 13.1: Service Overlay Network

En esta propuesta se asumirá que cada usuario de la red SON y los service gateways se conectarán a la o las redes de acceso correspondientes con algún acuerdo de servicio SLA con su proveedor que le permita tener ancho de banda suficiente hasta el backbone.

Los servicios básicos que requieran los usuarios, serán ofrecidos por los ISPs que ofrecen el acceso a Internet. Los usuarios podrán acceder a un conjunto de servicios garantizados que se ofrecerán a través de la “red overlay”.

Estos servicios podrían ser por ejemplo: video bajo demanda, juegos de alta calidad en línea, transmisión en vivo de eventos, servicios multimedia de e-learning, servicios multimedia de realidad virtual, video-conferencias, canales especiales de TV, etc..

Se asume que por estos servicios el usuario estará dispuesto a pagar un cierto precio adicional al precio de conectividad básica ofrecido por su ISP.

La arquitectura SON (intensiva en capital ya que el proveedor debe contratar las conexiones entre los services gateways), se debe justificar por la disposición a pagar de los usuarios. Esta disposición a pagar se dará en la medida que se pueda ofrecer un servicio que realmente asegure la calidad de servicio requerida para las aplicaciones correspondientes y por lo atractivo que puedan ser los servicios que ofrezcan estas aplicaciones.

Este tipo de servicios multidominio debería mejorar factores de escala de cada servicio (al permitir que cada servicio sea accedido por un conjunto de usuarios de múltiples dominios). Además, debería mejorar factores de escala de la propia red overlay permitiendo un uso más eficiente de la infraestructura overlay por múltiples servicios diferentes.

La posibilidad de que el usuario pueda recibir el servicio en múltiples redes de acceso debería ser también un factor a favor de afiliarse a este tipo de servicios. Si bien este tipo de servicios overlay puede ser ofrecido por terceros operadores OSPs (Overlay Service Providers), podría ser también ofrecido por un conjunto de ISPs que se asocien para tal fin o por un OSP consorciado con ISPs en diferentes zonas de Internet.

Se realizan para este caso de estudio, algunos agregados a la arquitectura general SON reseñada. Como se muestra en la figura 13.2, por un lado se hace explícita la posibilidad de que más de un service gateway se conecte a la misma red de acceso y también que un service gateway pueda estar conectado a más de una red de acceso. Esta posibilidad no estaba explícita en la arquitectura original. Sin embargo, facilita el acceso a los usuarios desde más de un service gateway aumentando la flexibilidad y facilitando de ser necesario el handover entre redes. Además, los servidores de las diferentes aplicaciones que se ofrezcan sobre la red overlay no se conectarán a las redes de acceso, sino directamente a los service gateway. De esta forma, se facilita su acceso al backbone overlay.

### 13.3. Redes triple play

El nombre triple play ha sido usado más como terminología del mundo comercial que del académico. Surge como una propuesta para los operadores tradicionales de telecomunicaciones. Los operadores tradicionales desean ofrecer en un único medio de acceso: voz (telefonía), video (Legacy TV, HDTV, PPV, VoD, video conferencias), HSI (de 1Mbps en adelante) y otros servicios como juegos en línea, etc..

El objetivo principal de una arquitectura triple play es ofrecer los servicios actuales y futuros de voz, datos y video en una red IP.

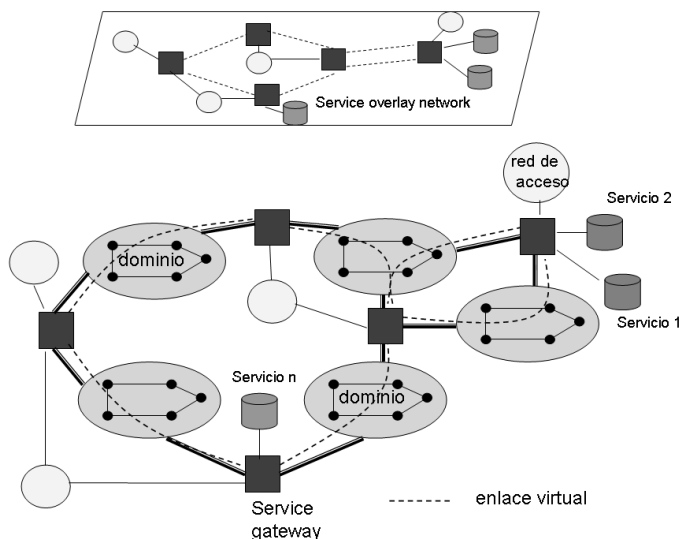


Figura 13.2: Modificaciones a la arquitectura SON

Lo que se conoce como arquitectura triple play no representa un quiebre importante con respecto a la arquitectura actual de las redes de los operadores, sino una evolución de las mismas. En la arquitectura triple play el paradigma de sobredimensionar el backbone sigue existiendo. Los cambios en el backbone pasan por dar algo más de “inteligencia” al mismo (por ejemplo utilizando MPLS y para hacer ingeniería de tráfico).

Donde se encuentran mayores cambios tecnológicos es en el acceso. Para dar estos servicios se requiere un acceso de mayor velocidad, con mayores funcionalidades y capacidades que los que ofrecían los equipos DSLAM y BRAS tradicionales.

Distintos proveedores ofrecen variaciones en cuanto a la arquitectura del acceso a la red. En general, los cambios se reflejan en una mayor capacidad en el acceso, el agregado de algunas funcionalidades como el multicast y con la posibilidad de realizar una reserva de ancho de banda para los servicios que tienen requerimientos de QoS.

En este tipo de redes se tiene una arquitectura similar a la actual con un backbone sobredimensionado, un acceso que es la limitante (pero con mayor ancho de banda y algunas funcionalidades adicionales) y la posibilidad de realizar reserva de recursos para ciertos servicios.

Un problema de los operadores que invierten en esta arquitectura es cómo obtener la mayor rentabilidad posible del acceso a estos nuevos servicios.

#### 13.4. Introducción a los mecanismos de subastas

En esta sección se realiza una breve introducción a los mecanismos de subastas. Un análisis más detallado de estos mecanismos y de su aplicación en el contexto de las redes puede verse en el libro de Courcoubetis y Weber [41].



Las subastas son mecanismos que se basan en la aplicación de dos reglas:

- la regla de asignación que define qué bienes se asignan a quiénes
- la regla de pago, que define cuánto pagan los ganadores de la subasta.

Las subastas se pueden clasificar en simples o de una unidad o multiunidad.

Pueden también ser de tipo abiertas o públicas (si las ofertas se hacen públicamente) o de tipo cerradas (si las ofertas se hacen en “sobre cerrado”).

Las subastas que maximizan la utilidad de los vendedores se dicen óptimas y aquellas que maximizan el bienestar social se dicen eficientes.

El pago de la subasta puede ser uniforme por todos los ganadores o discriminatorio. Dentro de las subastas simples hay varios mecanismos como por ejemplo:

- la subasta Inglesa donde se va aumentando el precio paulatinamente desde un precio base,
- la subasta Holandesa donde se va disminuyendo el precio desde un precio inicial máximo,

Ambas son de tipo público o abiertas. La subasta inglesa es quizás la que se está mas extendida en las subastas públicas habituales.

En el caso de subastas simples cerradas se presentan dos mecanismos las subastas de “primer precio” o de segundo precio (Vickrey). En estos casos, los oferentes envían sus ofertas cerradas y se asigna al mejor precio. En el caso de las de “primer precio” el ganador paga su oferta. En las subastas tipo Vickrey [144], el ganador paga el segundo mayor precio. Este último mecanismo de subasta, se prueba que tiene la propiedad de incentivar al usuario para revelar su verdadera utilidad en su oferta, esa es la estrategia dominante en el juego. Esta propiedad se denomina “incentive compatibility”.

En las subastas multi-unidad, cada oferente ofrece un par  $(p, q)$  de su voluntad de pagar hasta un precio  $p$  por una cantidad de  $q$  unidades. Todas las subastas simples pueden ser generalizadas a multi-unidad. Sin embargo, la única que mantiene la propiedad de “incentive compatibility” son las generalizaciones de las Vickrey.

Las subastas multiunidad de Vickrey (Vickrey generalizadas o también llamadas Vickrey-Clarke-Groves (VCG)) funcionan así:

- Cada oferente reporta su valuación de un subconjunto o de todos los puntos de su función de demanda.
- Las unidades son asignadas a las ofertas mayores hasta que se agota el recurso subastado.
- Cada oferente paga por la pérdida de beneficio social que el impone a los demás a través de su presencia.

Por ejemplo (ejemplo tomado de [49]), si se supone que 6 unidades son subastadas entre 5 oferentes (A,B,C,D,E) y que

- Los usuarios A y B están interesados en obtener exactamente 4 unidades

- Los usuarios C, D y E exactamente 2 unidades.
- Las ofertas son atómicas (todo o nada). El valor por unidad ofrecido por cada usuario es (9,8,6,7,2).

En este caso, A obtiene sus cuatro unidades y D obtiene dos unidades. En este caso, el precio que pagan es:

- A paga por la oferta más alta excluyendo la suya (ese es el costo social que el impone por participar de la subasta) es decir que paga: 4 unidades a 8 pesos la unidad.
- D paga 2 unidades a 6 pesos la unidad. Este es el costo social que D impone al participar ya que quien ganaría en la subasta si D no participara sería C (B igual no ganaría porque solicitó 4 unidades y hay 2 disponibles).

La subastas generalizadas de Vickrey (VCG) se prueba que verifica tres importantes propiedades:

- 1) Incentiva a que el usuario revele su verdadera utilidad en la oferta
- 2) Racionalidad individual: cada jugador obtiene una utilidad no negativa
- 3) Eficiencia, el beneficio social es maximizado.

### 13.5. Planteo del problema y análisis de subastas en un enlace

La arquitectura SON y la arquitectura triple play son dos propuestas que buscan sobre la arquitectura actual de Internet ofrecer una red overlay de servicios premium con requerimientos de calidad de servicio. En ambos casos el objetivo del ISP o del OSP es optimizar el uso de su recurso escaso, procurando obtener la mayor rentabilidad por su venta y controlar el acceso para asegurar calidad de servicio.

Si bien el recurso escaso en la arquitectura triple play es principalmente el acceso y en la arquitectura SON es el backbone, el problema general de controlar el acceso a una red de servicios premium, en la que los usuarios compartan los recursos de la red y se maximice la ganancia del operador es el mismo en ambos casos. La inversión en ambos, se justifica si existen usuarios dispuestos a pagar por estos servicios de valor agregado. Es necesario asegurar a quien accede a la red la disponibilidad de los recursos necesarios mientras dure su conexión. Por último, en ambos casos el desafío es obtener del conjunto de usuarios que desean usar estos servicios la mayor ganancia para el operador.

En la propuesta de este trabajo, los usuarios ganarán acceso a la red en la medida que:

- Existan en la red los recursos necesarios para asegurarle la calidad requerida para el servicio solicitado.
- Su oferta por el servicio “gane” en una subasta con los demás usuarios que compiten por el recurso escaso correspondiente.

Cada servicio premium dispondrá de un agente intermediario (o broker) que administrará de forma distribuida con los restantes agentes el acceso a la red de los usuarios. Cada agente administrará un conjunto de rutas para acceder al servicio correspondiente. La ruta  $r$  estará asociada a un par de nodos de ingreso/egreso y al servicio correspondiente.

La red estará compuesta de un conjunto de enlaces indexados por  $l$ , y un conjunto de rutas indexadas por  $r$ .  $R$  será la matriz de ruteo,  $R_{lr} = 1$  si la ruta  $r$  incluye al enlace  $l$ , en otro caso  $R_{lr} = 0$ .  $c = (c_l)$  es el vector de las capacidades de los enlaces. Como se dijo antes, a cada ruta  $r$  está asociada un servicio con un requerimiento de ancho de banda  $\sigma_r$ : los usuarios realizarán ofertas por este servicio con este requerimiento de ancho de banda necesario para asegurar sus requerimientos de QoS. Pueden haber servicios diferentes con requerimientos de QoS diferentes sobre un mismo camino físico; en este caso se usará un índice diferente  $r$  para cada servicio diferente, por lo que esta formulación es totalmente general. Para cada ruta  $r$ , la red recibe un conjunto  $N_r$  de ofertas  $b_r^{(i)}$ , que son ordenadas de la siguiente forma:

$$b_r^{(1)} \geq b_r^{(2)} \geq \dots \geq b_r^{(N_r)}.$$

La decisión a tomar es cuáles de estas ofertas aceptar y asignarles recursos en la red, respetando las restricciones de capacidad, a los efectos de maximizar la ganancia del operador. Se asumirá una subasta tipo “primer precio”, donde los usuarios que acceden a la red pagarán por su oferta. No se utilizará en este caso el mecanismo de VCG que es el que habitualmente utilizan en la literatura de subastas en redes para obtener que el usuario revele su verdadera utilidad y maximizar el beneficio social. En el capítulo siguiente en la sección 14.6 se discutirá este punto con más detalle y se fundamentará la elección del mecanismo de primer precio.

Se analizará primero el caso del acceso a un servicio con un solo enlace. Este caso permitirá ir agregando complejidad paso a paso al problema. Además este problema tiene interés en sí mismo ya que por ejemplo en el modelo donde el backbone está realmente sobredimensionado, la competencia es entre los usuarios que comparten el ancho de banda del enlace de acceso a la red.

Las subastas se realizan periódicamente entre las ofertas recibidas durante un período de tiempo  $T$ . Una vez que los recursos se asignan, esta asignación se mantiene durante toda la duración del servicio que típicamente excede el tiempo  $T$ . En este esquema no se permite que las ofertas desplacen a las conexiones ya establecidas en períodos anteriores.

Se busca una política de asignación de recursos que maximice la ganancia del operador en el tiempo. En esta primera sección se estudiará el caso de un único enlace de capacidad  $C$ , con una sola clase de servicio con requerimiento de capacidad  $\sigma = 1$ .

El índice de tiempo  $k$  define la subasta en el tiempo  $kT$ , que involucra el conjunto de ofertas ordenadas  $b^{k,(i)}$ . Si  $a^k$  representa la tasa de información admitida (en este caso igual al número de conexiones admitidas  $m^k$  ya que  $\sigma = 1$ ), la ganancia asociada es:

$$U_{b^k}(a^k) := \sum_{i=1}^{a^k} b^{k,(i)} \quad (13.1)$$

$U_{b^k}(\cdot)$  puede ser interpolada para definir la función creciente, lineal a tramos y cóncava sobre los reales. Esta extensión se puede hacer definiendo  $U$  constante para tasas mayores que  $a^k$ . En la figura 13.3 se muestra un caso típico de esta función.

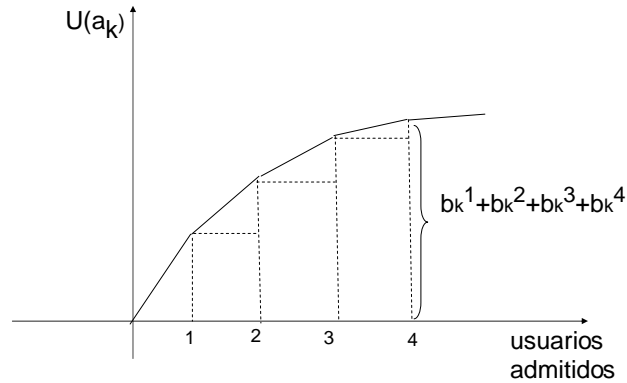


Figura 13.3: Ganancia del operador en la subasta  $k$

### 13.5.1. Asignación óptima como un Markov Decision Process

El problema de optimizar la ganancia a largo plazo del operador se puede representar con un modelo estocástico para las ofertas y para la duración del servicio. Se asumirá que las ofertas son sorteadas de una cierta distribución de probabilidad, y que los tiempos de servicio son modelados como variables aleatorias exponenciales independientes de media  $1/\mu$ . Por lo tanto, al final del período de duración  $T$  cada conexión tiene probabilidad  $p := e^{-\mu T}$  de permanecer activa para el período siguiente.

**Observación 13.1.** *Se asume en esta sección que la distribución de las ofertas es conocida por el rematador. También se asume que el número de ofertas en cada subasta es fijo y conocido  $N$ . En el siguiente capítulo en la sección 14.5 se considerará el caso en que la distribución de las ofertas se aprende de la observación de las mismas en períodos anteriores. También se analizará la posibilidad de que las ofertas arriben como un proceso aleatorio.*

**Observación 13.2.** *La hipótesis más cuestionable es que la duración del tiempo de servicio sea exponencial. La duración es una característica del servicio que se remata y en algunos casos podría ser considerado aleatorio y en otros como determinístico. En muchos casos, la única aleatoriedad es que el usuario termine el servicio antes de que termine el tiempo del servicio (por ejemplo antes de que finalice la película). Claramente, una distribución exponencial no captura esto bien, pero ha sido utilizado para adoptar un modelo markoviano y poder analizar el problema. El parámetro correspondiente a la duración  $\mu$  está asociado con el servicio que se está rematando, por lo tanto en esta sección es el mismo para todos los usuarios; en la sección 14.4 se puede ver como se generaliza este caso.*

Dada la distribución de las ofertas  $b$ , se define la función de ganancia esperada

$$\bar{U}(a) = E[U_b(a)] \quad (13.2)$$

donde se reemplazan las ofertas recibidas en esta subasta (13.1) por su valor esperado. Como las ofertas están ordenadas, en el lugar de cada oferta en la suma de la ecuación 13.1, aparecen como sumandos el valor esperado del estadístico de orden correspondiente. Esta función también es creciente, lineal a tramos y cóncava.

Sea  $x^k$  el número de conexiones activas en  $t = kT^-$ , es decir antes de la  $k$ -ésima subasta. El sistema admite  $a^k$  conexiones nuevas,  $0 \leq a^k \leq C - x^k$ , llevando el total de conexiones a  $x^k + a^k$ . Al final de dicho período,  $t = (k+1)T^-$ , el número de conexiones activas  $x^{k+1}$  es una variable aleatoria con distribución binomial con parámetros  $x^k + a^k$  y  $p$ :

$$P[x^{k+1} = i | x^k, a^k] = \binom{x^k + a^k}{i} p^i (1-p)^{x^k + a^k - i}. \quad (13.3)$$

En la figura 13.4 se representa la dinámica descrita para las subastas periódicas.

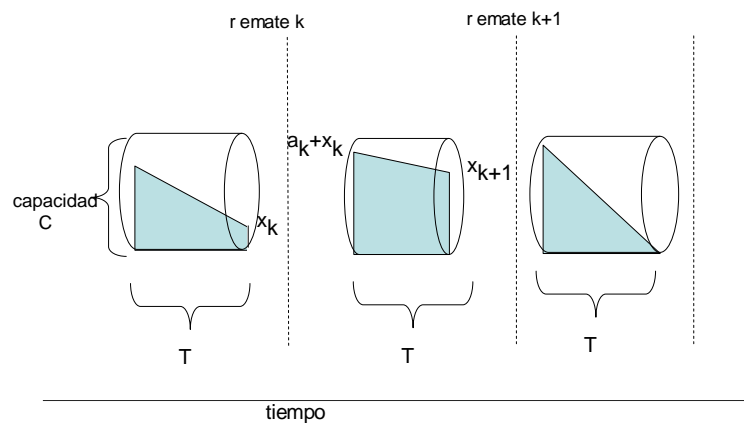


Figura 13.4: Dinámica de las subastas en un enlace

**Problema 13.1 (Optimizar la ganancia media, un enlace).**

$$\text{Maximizar } \lim_n \frac{1}{n} \sum_{k=0}^{n-1} E[U_{b^k}(a^k)].$$

En esta expresión, el valor esperado tiene dos fuentes de aleatoriedad: el vector de las ofertas  $b^k$  y el proceso de partidas. Las restricciones son  $0 \leq a^k \leq C - x^k$  donde  $x^k$  sigue una dinámica de

transiciones de acuerdo a la distribución binomial (13.3). También se puede considerar la versión con descuento:

$$\text{Maximizar } \sum_{k=0}^{\infty} \rho^k E[U_{b^k}(a^k)], \text{ donde } 0 < \rho < 1.$$

Ambas formulaciones corresponden a Markov Decision Processes (MDP). Un análisis detallado de Markov Decision Processes se puede ver en el libro de Bertsekas [20] o en el libro de Puterman [123].

Un MDP está definido a partir de:

- Un espacio de estados  $\mathcal{S}$ .
- Un conjunto de posibles acciones a tomar en cada estado  $\mathcal{A}_s$ .
- Una función  $U(a^k, s^k)$  que representa la ganancia que se obtiene cuando el sistema está en el estado  $s^k$  y se toma la acción  $a^k$ .
- Un conjunto de probabilidades  $p(s^{k+1}, s^k, a^k)$  que indican la probabilidad de que estando en el estado  $s^k$  y aplicando la acción  $a^k$ , el sistema pase en el próximo tiempo a estar en el estado  $s^{k+1}$ .
- Una función de ganancia a maximizar  $V$  (o de costo a minimizar) si el sistema parte del estado  $s^0$ :
  - Si el problema es de utilidad descontada (siendo  $\rho$  el factor de descuento):

$$V(s^0) = \lim_{N \rightarrow \infty} \mathbf{E}\left\{ \sum_{k=0}^{N-1} \rho^k U(a^k, s^k) \right\} \quad (13.4)$$

- Si el problema es de utilidad promedio:

$$V(s^0) = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{E}\left\{ \sum_{k=0}^{N-1} U(a^k, s^k) \right\} \quad (13.5)$$

En el MDP asociado a las subastas en un enlace, el *estado* en el tiempo  $k$  viene dado por  $s^k = (x^k, b^k)$ , esto es la ocupación actual del enlace y las ofertas que arribaron en esta subasta. Basados en este estado, la *acción*  $a^k = a(s^k)$  decide cuántas ofertas aceptar. Una solución a un MDP es encontrar una *política*  $a(s)$  que minimice un costo (o maximice una ganancia en nuestro caso). Una política es un conjunto de acciones (una por estado del sistema). En la teoría de MDP se prueba que la política óptima es estacionaria. Esto significa para el problema formulado, que dada la cantidad de circuitos ocupados y el valor de las ofertas que arribaron en esta subasta, existe una acción (que indica cuántas de esas ofertas dejar ingresar) que es óptima. En el caso con descuento  $\rho < 1$ , esta política satisface la ecuación de Bellman

$$V^*(x^0, b) = \max_{a \in \mathcal{A}_s} \{U_b(a) + \rho E[V^*(x^1, b')]\}, \quad (13.6)$$

donde  $V^*$  es la función de valor óptima y el valor esperado se toma sobre la distribución binomial de  $x^1 | (x^0, a)$  y la distribución de la próxima oferta  $b'$ . Las restricciones de este problema de optimización dependen del estado y son  $\mathcal{A}_s = \{0 \leq a \leq C - x^0\}$ . Cuando  $\rho = 1$ , el  $V^*$  que satisface la ecuación (13.6) no es más la ganancia óptima, pero la ecuación (13.6) aún caracteriza la política óptima  $a(s)$ .

En general es difícil resolver la ecuación de Bellman; un algoritmo habitualmente utilizado para este fin en MDPs es el algoritmo conocido como “*value iteration*”

$$V_{m+1}(x^0, b) := \max_{a \in \mathcal{A}_s} \{U_b(a) + \rho E[V_m(x^1, b')]\};$$

comenzando con un valor arbitrario de  $V_0(s)$ ,  $V_m(s)$  converge a  $V^*(s)$ , y la acción correspondiente converge a la acción óptima [20].

El MDP anterior presenta como dificultad principal para su solución, la dimensión del estado. Esto se debe a que en el estado intervienen las ofertas. El espacio de estados es un espacio continuo por este motivo. Pero aún cuando se discretice el valor de las ofertas, se tiene una cantidad de estados igual al producto del número de posibles circuitos ocupados por el número de posibles valores de las ofertas por  $N$  el número de ofertas. Aún para un enlace con capacidad  $C$  para pocos circuitos este número puede ser grande.

La resolución de este MDP incluso por métodos numéricos involucra un costo computacional alto. Parece aún más difícil lograr una solución analítica de este problema. No se debe olvidar que un objetivo final es poder abordar el problema de la red. En el caso de la red el estado es aún más intratable ya que pasa a intervenir en el estado el número de enlaces de la red y también aparecen ofertas por diferentes caminos. Por este motivo se buscará algún método que aproxime la solución al problema de un enlace y que contribuya a entender el problema de la red y a encontrar un método distribuido que permita resolverlo.

Antes de ver la solución propuesta al MDP formulado, para comprender mejor el problema y ganar en intuición sobre su posible solución, se analizará a continuación una versión simplificada del mismo. Posteriormente en la sección 13.6 se verá la solución propuesta al MDP formulado en esta sección.

## Modelo Simplificado

En este modelo simplificado, el valor actual de las ofertas no participa en el estado del MDP. Este caso se puede pensar como el anterior pero donde las ofertas tienen un valor constante en cada remate e igual a su valor esperado. El estado está definido sólo por los circuitos ocupados antes de cada subasta y por lo tanto la cantidad de ingresos en cada subasta es sólo función de esa variable.

El valor actual de la oferta, sólo se utiliza una vez resuelto el MDP y encontrada la cantidad óptima de usuarios que deben ingresar, para decidir quiénes son los usuarios que más ofrecieron en este remate. En esta sección se analizará cuántos usuarios dejar ingresar en cada estado tomando

como parámetros la duración media del servicio (a través de la probabilidad  $p$ ) y el valor esperado de las ofertas. La cantidad de usuarios que ingresan en cada estado dependerá de la relación entre la duración media del servicio y el valor medio de las ofertas. Por ejemplo, si se tienen dos circuitos libres y la segunda oferta en media es muy baja comparada con el valor medio de la primera oferta y/o el servicio tiene una duración media muy alta, es probable que convenga dejar uno de los circuitos libres. Si por el contrario, el servicio finaliza con probabilidad 1 en el período entre dos remates, siempre conviene ocupar todos los circuitos libres, porque en la subasta siguiente volverán a estar todos libres.

Esta versión simplificada al manejar un espacio de estados más simple, permitirá hacer algunos cálculos analíticos y también permitirá resolver por métodos numéricos en tiempos razonables casos donde no se puede obtener una solución analítica. Este análisis permitirá ganar intuición sobre el problema, su complejidad y en cómo analizar el problema cuando las ofertas son parte del estado del sistema.

La ecuación de Bellman pasa ahora a tener la siguiente forma más simple:

$$V^*(x^0) = \max_{a \in \mathcal{A}_s} \{ \bar{U}(a) + \rho E[V^*(x^1)] \}, \quad (13.7)$$

Se analizarán a continuación algunos ejemplos con este modelo simplificado

### **Caso 1: MDP en enlace con capacidad para una conexión**

El ejemplo más simple es considerar un enlace que tiene capacidad para una única conexión. En el modelo simplificado el estado es 0 (si el enlace está vacío en el instante en que se remata) o 1 (si el enlace está ocupado).

Existe una única política estacionaria de interés que es dejar ingresar una conexión nueva cuando el estado indica que el enlace está vacío. Asumimos que las conexiones son independientes y tienen probabilidad  $p$  de permanecer activas entre dos subastas. Es decir que cuando hay que rematar, si el enlace está vacío se deja ingresar un servicio y se tiene probabilidad  $1 - p$  de que dicha conexión termine en el período hasta la próxima subasta y  $p$  de que en la próxima subasta el enlace continúe ocupado. Cuando está ocupado al momento de rematar no se deja ingresar usuarios y hay probabilidad  $1 - p$  de que en la próxima subasta esté vacío el enlace. En la figura 13.5 se muestra la cadena de Markov asociada a esta única política de interés.

La solución estacionaria de la cadena de Markov correspondiente a esta política es:

$$\begin{aligned} \pi_0 &= 1 - p \\ \pi_1 &= p \end{aligned}$$

siendo  $\pi_0$  la probabilidad del estado estacionario de tener el enlace sin ninguna conexión establecida y  $\pi_1$  la probabilidad del estado estacionario de tener una conexión establecida.

En este caso se puede calcular la ganancia del sistema a partir de la solución estacionaria anterior. El valor esperado de lo que ganará el sistema es la probabilidad estacionaria de tener un circuito libre por el valor esperado de la mejor oferta, es decir:  $V = (1 - p)E(b^{(1)})$ , siendo  $N$  la cantidad de ofertas que se reciben en cada subasta y  $b^{(1)}$  la variable aleatoria que representa la mayor de las  $N$  ofertas recibidas.



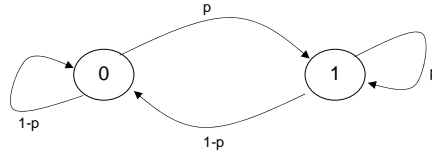


Figura 13.5: Cadena de Markov, Caso 1

Por ejemplo, si la distribución de las ofertas es uniforme  $[0,1]$ , entonces

$$E(b^{(1)}) = \frac{N}{N+1}$$

y por lo tanto la ganancia media será:

$$V = (1-p) \frac{N}{N+1}$$

En general, si las ofertas tienen una distribución de soporte acotado y  $b_{max}$  es el extremo derecho de dicho soporte, si  $N$  es grande entonces

$$E(b^{(1)}) \approx b_{max}$$

y por tanto la ganancia será

$$V \approx (1-p)b_{max}.$$

### Caso 2: MDP en enlace con capacidad para dos conexiones

Este caso también es suficientemente simple para poder analizarlo analíticamente. En este caso hay dos políticas que interesa comparar:

- Política 1. La primera política siempre llena el enlace. Es decir que si hay lugar para una conexión se deja ingresar la mayor oferta recibida y si hay lugar para dos ofertas se dejan ingresar las dos mejores ofertas.
- Política 2. La segunda política difiere de la primera sólo en que si hay lugar para dos conexiones, únicamente se deja ingresar a la red una conexión, la correspondiente a la mejor oferta. La idea es no llenar el enlace porque es probable que en la próxima subasta se tenga una oferta mejor que la segunda recibida y que no se dejó ingresar.

En ambos casos cada política genera una cadena de Markov diferente pero que puede ser resuelta analíticamente.

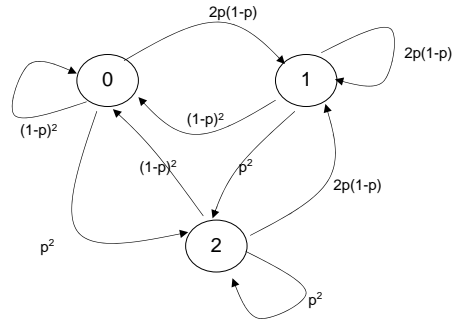


Figura 13.6: Cadena de Markov, Política 1

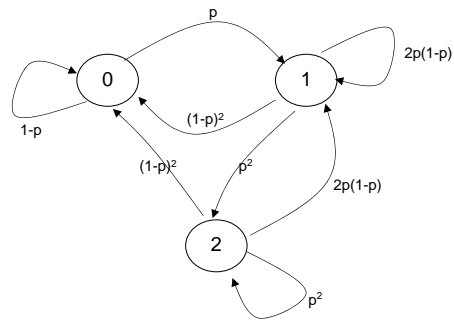


Figura 13.7: Cadena de Markov, Política 2

Se puede calcular la probabilidad de estado estacionario para cada cadena y se obtiene:  
Política 1:

$$\begin{aligned}\pi_0 &= (1-p)^2 \\ \pi_1 &= 2p(1-p) \\ \pi_2 &= 1 - \pi_1 - \pi_0\end{aligned}$$

Política 2:

$$\pi_0 = \frac{(1-p)^2}{p^2 - p + 1}$$

$$\pi_1 = \frac{p(1-p)^2}{p^2 - p + 1}$$

$$\pi_2 = 1 - \pi_1 - \pi_0$$

siendo  $\pi_0$  la probabilidad estacionaria de que no exista ninguna conexión establecida.

Sea  $E(b^{(1)})$  el valor esperado de la oferta máxima y  $E(b^{(2)})$  el valor esperado de la segunda mejor oferta. El valor esperado de la ganancia en cada caso es:

$$\text{Política 1 : } V_1 = \pi_0(E(b^{(1)}) + E(b^{(2)})) + \pi_1 E(b^{(1)})$$

$$\text{Política 2 : } V_2 = E(b^{(1)})(\pi_1 + \pi_0)$$

Si sustituimos las probabilidades estacionarias y evaluamos cuando  $V_2 > V_1$ , es decir cuando conviene no llenar el enlace y esperar a la próxima subasta se llega a que:

$$\frac{E(b^{(2)})}{E(b^{(1)})} < \frac{p^2}{p^2 - p + 1} \quad (13.8)$$

En la figura 13.8, se muestran dos zonas separadas por la curva  $\frac{E(b^{(2)})}{E(b^{(1)})} = \frac{p^2}{p^2 - p + 1}$ . En la zona que se encuentra por encima de dicha curva es conveniente utilizar la política que siempre llena el enlace (política 1). En la zona inferior es conveniente utilizar la política 2. Para el caso de ofertas con distribución uniforme, en la figura 13.9 la curva indica para una probabilidad  $p$  dada el mínimo  $N$  para el cual conviene no llenar el enlace y usar la política 2.

Cuando  $p$  es grande, los circuitos se liberan poco y por tanto cuando se liberan los dos puede convenir llenar sólo uno (no llenar el segundo con la segunda mejor oferta), y esperar a la próxima subasta para llenar el otro libre con la mejor oferta de la próxima subasta. Cuando  $p$  es pequeño siempre conviene llenar los dos circuitos (política 1). El valor cuando deja de convenir una política y pasa a convenir la otra depende de  $N$ . En la figura 13.10 se muestra cuál es la política óptima cuando hay dos circuitos libres en función de  $p$  y para diferentes valores de  $N$ . Cada línea vertical indica que a su derecha conviene utilizar la política 2 y a su izquierda la política 1. Cada línea vertical es el umbral de transición para los diferentes valores de  $N$ .

### Caso 3: MDP en un enlace con capacidad para $K$ conexiones

La siguiente pregunta es qué sucede si tengo más circuitos. En estos casos es más difícil obtener un resultado analítico pero es posible resolver numéricamente el problema y analizar si el comportamiento visto para dos circuitos es generalizable y qué diferencias existen a medida que aumenta el número de éstos.

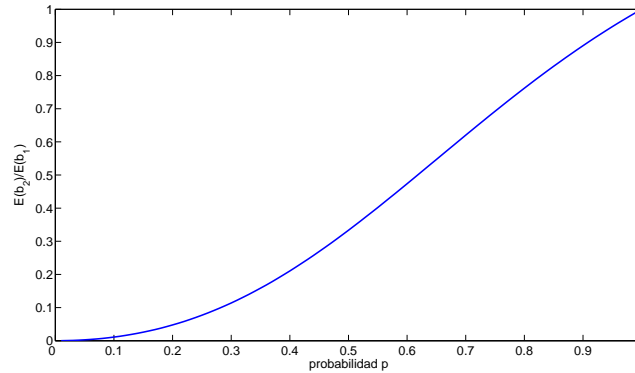


Figura 13.8: Zonas donde conviene utilizar la política 1 o 2

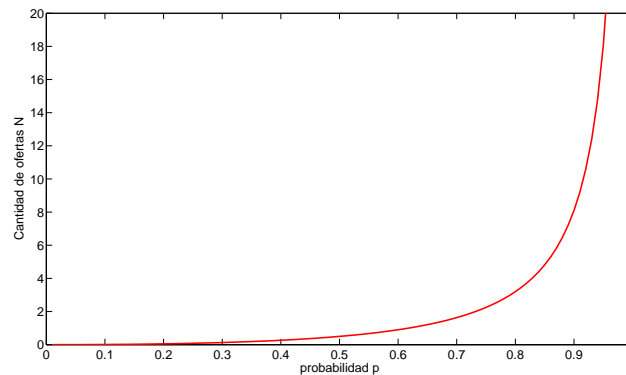


Figura 13.9:  $N$ , número de oferentes mínimo en función de  $p$

Por ejemplo si en lugar de 2 circuitos se tienen 5 circuitos en la capacidad del enlace, en la figura 13.11 se puede ver la política óptima cuando se vacían los 5 circuitos si  $N = 5$ . En esta figura se observa que cuando la probabilidad de que la conexión termine entre dos subastas conviene ocupar los 5 circuitos que quedaron libres. A medida que esta probabilidad disminuye cada vez conviene ocupar menos circuitos y dejar libres para ocuparlos con mejores ofertas en media en la próxima subasta. Se puede calcular también la política óptima para cada estado en función de  $p$ . Se representa la política óptima para un valor de  $p$  por un vector de 6 componentes  $[x_5, x_4, x_3, x_2, x_1, x_0]$  donde  $x_i$  representa la cantidad de ofertas que se deben dejar ingresar cuando hay  $i$  circuitos libres en el momento de rematar. Con esta representación, la política óptima para  $C = 5$  y  $N = 5$  para cada valor de  $p$  se puede ver en el cuadro 13.1.

#### MDP con conexiones de diferente duración media

Por último se analiza el caso de tener dos servicios diferentes al que quieren acceder los usuarios a través del enlace, y donde cada servicio tiene diferente duración media. Las ofertas por ambos servicios tienen distribución uniforme en  $[0,1]$ .

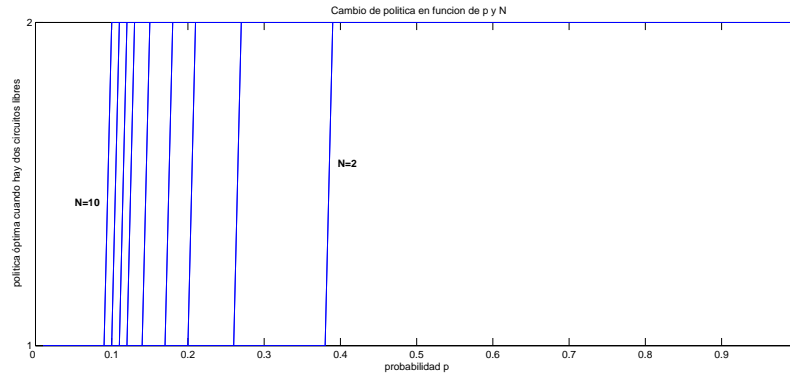


Figura 13.10: Política óptima en función de p

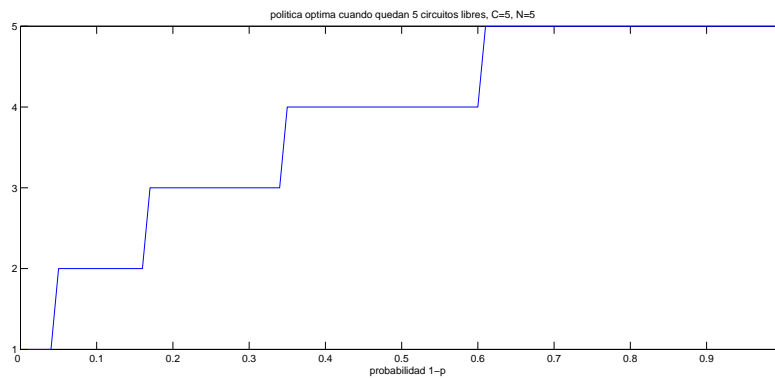


Figura 13.11: Política óptima, cantidad de circuitos que se deben ocupar en función de la probabilidad de muerte  $1-p$

En la figura 13.12 se analiza cuál es la política óptima para el caso de dos fuentes en función de  $q_1 = 1 - p_1$  y  $q_2 = 1 - p_2$  (probabilidades de que termine una conexión entre el tiempo de dos subastas para cada servicio).

En la figura 13.13, se muestra la misma comparación pero para el caso de  $N = 10$ . Es claro que cuando  $N$  se hace muy grande las dos mejores ofertas de cada servicio tienden a 1 y por lo tanto para  $N$  muy grande simplemente conviene dejar ingresar a las ofertas del servicio con menor tiempo de conexión medio.

Cuadro 13.1: Política óptima para cada  $p$ 

probabilidad	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$
0.95	1	1	1	1	1	0
0.9	2	2	2	1	1	0
0.8	3	3	2	2	1	0
0.7	3	3	3	2	1	0
0.6	4	3	3	2	1	0
0.5	4	4	3	2	1	0
0.4	4	4	3	2	1	0
0.3	5	4	3	2	1	0
0.2	5	4	3	2	1	0
0.1	5	4	3	2	1	0
0	5	4	3	2	1	0

### 13.6. Aproximación mediante la reducción del horizonte

Se analizará ahora el problema que considera en el estado el valor de la oferta actual. Esto es importante porque en el modelo simplificado anterior la única consideración para aceptar o no una oferta en cierto estado es la relación entre los valores esperados de los estadísticos de orden de la distribución de las ofertas y la probabilidad  $p$  asociada a la duración media del servicio. Al no considerar el valor de la oferta actual en el estado, podría suceder que por la relación entre los estadísticos de orden, habiendo dos circuitos libres se rechace la segunda oferta y sin embargo en una subasta particular la segunda oferta sea muy buena. También podría suceder lo inverso, que por los valores medios se decidiera aceptar las dos primeras ofertas cuando quedan dos circuitos libres y en una subasta particular, la segunda oferta tuviera un valor muy inferior al medio y no convenga aceptarla.

El modelo no simplificado presenta una mayor dificultad analítica y también en costo computacional para resolverlo numéricamente. Aún en el caso del modelo simplificado es muy difícil llegar a resultados analíticos salvo en casos muy simples como los que se vieron de enlaces con capacidad para uno o dos circuitos. En el modelo no simplificado salvo en casos muy simples (de enlaces con conexiones para pocos circuitos) la solución numérica utilizando el algoritmo de “value iteration” por ejemplo tiene un costo computacional muy alto. No se debe perder de vista además que el objetivo es resolver el problema de una red, donde obviamente el estado tiene una representación más compleja aún.

Por este motivo, se buscará una solución aproximada y se analizará cómo se comporta esta solución en casos simples donde se puede encontrar la solución óptima al MDP. Además esta aproximación aportará las ideas principales para buscar una solución al caso de la red que se estudia en el capítulo siguiente.

Para buscar esta solución se analizarán los primeros pasos del algoritmo de value iteration y de

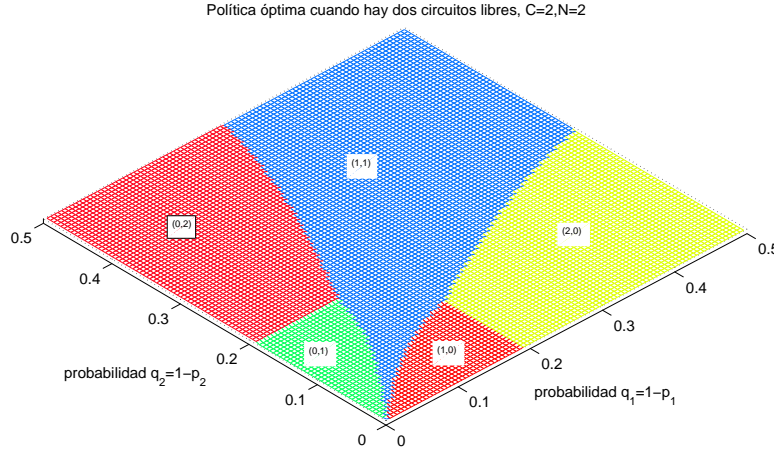


Figura 13.12: Política óptima para el caso de dos servicios, con  $C = 2$ ,  $N = 2$

esta forma se aproximará la política óptima. Comenzando con  $V_0 \equiv 0$ , se tiene que

$$V_1(x^0, b) = \max_{a \leq C - x^0} U_b(a) = U_b(C - x^0).$$

El primer paso brinda lo que se llamará la política “miope”  $a = C - x^0$ , que vende toda la capacidad disponible sin tener en cuenta el futuro. En ciertos escenarios esta puede ser una buena política. Por ejemplo cuando la cantidad de ofertas es muy grande comparado con la cantidad de circuitos, la política miope es suficientemente buena ya que es muy probable en este caso que siempre se reciban muy buenas ofertas y por lo tanto siempre convenga llenar los circuitos.

Para mejorar la política miope, se considera un segundo paso del algoritmo de value iteration:

$$\begin{aligned} V_2(x^0, b) &= \max_{a \leq C - x^0} \{U_b(a) + \rho E[V_1(x^1, b')]\} \\ &= \max_{a \leq C - x^0} \{U_b(a) + \rho E[U_{b'}(C - x^1)]\} \\ &= \max_{a \leq C - x^0} \{U_b(a) + \rho E_{x^1} \bar{U}(C - x^1)\}. \end{aligned} \quad (13.9)$$

En (13.9), se tomó el valor esperado con respecto a las ofertas  $b'$ , usando la función  $\bar{U}$  definida antes; lo que resta es el valor esperado con respecto a  $x^1 \sim \text{Bin}(x^0 + a, p)$ . La política que resuelve (13.9) puede ser interpretada como una *reducción del horizonte*: la decisión optimiza sobre la ganancia actual más el valor esperado de la ganancia mirando un paso hacia el futuro, asumiendo que toda la capacidad disponible será vendida en el próximo paso. Esta decisión es aplicada recursivamente y por lo tanto el futuro es tomado en cuenta, pero sólo en un nivel limitado de complejidad.

El primer término en (13.9) crece con  $a$ . Para caracterizar el segundo término, se puede reescribirlo de la siguiente forma. Se considera la función  $W(i) = \bar{U}(C) - \bar{U}(C - i)$ , lineal a tramos,

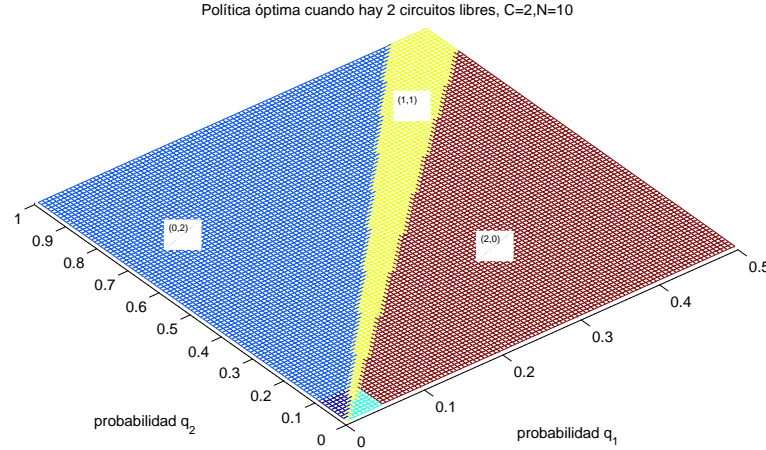


Figura 13.13: Política óptima para el caso de dos servicios, con  $C = 2, N = 10$

creciente y *convexa* en  $i$ . Los incrementos tendrán la forma:

$$w(i) := W(i) - W(i-1) = E[b^{(C-i)}], \quad i = 1, \dots, C$$

y son no negativos y crecientes en  $i$  (dado que las ofertas están ordenadas de forma decreciente). Se analizará ahora el valor esperado con respecto a la distribución binomial.

**Proposición 13.1.** *Se define  $\bar{W}(x) = E[W(I_x)]$ , donde  $I_x \sim \text{Bin}(x, p)$  para valores enteros de  $x$ , y se extiende por interpolación lineal. Entonces  $\bar{W}(x)$  es creciente y convexa.*

*Demostración.* Dado  $I_x \sim \text{Bin}(x, p)$ , con  $x$  entero, se puede generar una variable aleatoria  $\text{Bin}(x+1, p)$  de la forma  $I_x + \xi$ , donde  $\xi$  es Bernoulli ( $p$ ), independiente de  $I_x$ . Escribiendo

$$W(I_x + \xi) - W(I_x) = w(I_x + 1)\xi$$

y tomando valor esperado, usando la independencia se obtienen incrementos

$$\bar{w}(x+1) := \bar{W}(x+1) - \bar{W}(x) = pE[w(I_x + 1)]. \quad (13.10)$$

Resta probar que el último término es creciente en  $x$ . Notando que  $w(i)$  es creciente, la desigualdad  $w(I_x + \xi + 1) \geq w(I_x + 1)$  es cierta casi seguramente y tomando valor esperado,

$$E[w(I_{(x+1)} + 1)] = E[w(I_x + \xi + 1)] \geq E[w(I_x + 1)].$$

□

La optimización mirando un paso hacia el futuro (13.9) puede ser reescrita entonces

$$\max_{a \leq C-x^0} U_b(a) - \rho \bar{W}(x^0 + a) + \rho \bar{U}(C). \quad (13.11)$$



Se encuentra implícito en (13.9) y (13.11) que  $a$  es un entero. En este caso, sin embargo, la condición puede ser relajada sin pérdida de generalidad, considerando entonces a (13.11) como un problema de optimización convexa. Para resolverlo se debe buscar el punto de cruce de las derivadas de  $U_b(a)$  y  $\rho\bar{W}(x^0 + a)$  (la utilidad y el costo marginal), como se muestra en la Fig. 13.14.

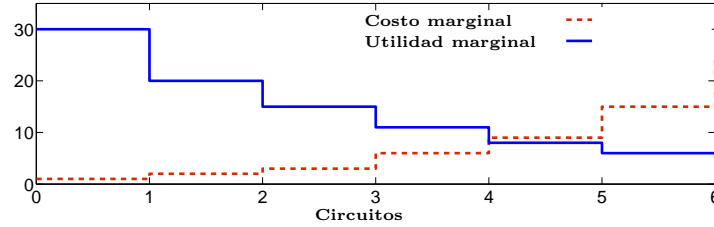


Figura 13.14: Utilidad marginal versus costo marginal

La utilidad marginal es el valor de las ofertas actuales en orden decreciente. El costo marginal representa el valor de dejar uno o más circuitos libres para la próxima subasta, y tiene la forma  $\rho\bar{w}(i)$ , con  $\bar{w}(\cdot)$  definido en (13.10); que como se mencionó es creciente en  $i$ . Dado que las ofertas son aleatorias  $b$ , las curvas de la figura 13.14 casi seguramente se cruzarán en un sólo punto y de valor entero. Por lo tanto en este caso la relajación convexa del problema entero no cambia la solución.

La política óptima de aceptación  $a$  es tal que

$$b^{(1)} \geq \dots \geq b^{(a)} \geq \rho\bar{w}(i) > b^{(a+1)}, \quad \text{para } i = x^0 + a.$$

Los valores  $\rho\bar{w}(i)$  actúan como *umbrales* sucesivos: para aceptar  $a$  ofertas, la *menor* debe exceder  $\rho\bar{w}(x^0 + a)$ . Para aceptar una más, se requiere un umbral *más exigente*  $\rho\bar{w}(x^0 + a + 1)$  en esta oferta menor.

Es interesante observar que con esta aproximación se puede calcular una fórmula explícita para los umbrales.

**Proposición 13.2.** *Conociendo los estadísticos de orden de la distribución de la ofertas y la probabilidad  $p$ , los umbrales se pueden calcular según la siguiente ecuación:*

$$\bar{w}(i) = p \sum_{l=0}^{i-1} E(b^{(C-l)}) \binom{i-1}{l} p^l (1-p)^{i-1-l}. \quad (13.12)$$

*Demostración.* De la ecuación (13.10) se tiene que  $\bar{w}(x) = E(W(I_x)) - E(W(I_{x-1}))$  con  $I_x \sim \text{Bin}(x, p)$ . De las definiciones de  $W$  y  $\bar{U}$  se puede reescribir la expresión anterior de la siguiente forma:

$$\begin{aligned} \bar{w}(x) &= E_{I_{x-1}} \left( \sum_{l=0}^{C-I_{x-1}} E(b^l) \right) - E_{I_x} \left( \sum_{l=0}^{C-I_x} E(b^l) \right) \\ &= \sum_{j=0}^{x-1} A(x-1, j) \sum_{l=1}^{C-j} E(b^l) - \sum_{j=0}^x A(x, j) \sum_{l=1}^{C-j} E(b^l), \end{aligned}$$

donde  $A(x, j) := \binom{x}{j} p^j (1-p)^{x-j}$ , operando se obtiene

$$\begin{aligned} \bar{w}(x) &= \sum_{l=C-x+1}^C E(b^l) \left( \sum_{j=0}^{C-l} A(x-1, j) - \sum_{j=0}^{C-l} A(x, j) \right) \\ &= \sum_{l=0}^{x-1} E(b^{C-l}) \left( \sum_{j=0}^l A(x-1, j) - \sum_{j=0}^l A(x, j) \right). \end{aligned} \quad (13.13)$$

Se puede probar por inducción que ,  $\sum_{j=0}^l A(x-1, j) - \sum_{j=0}^l A(x, j) = pA(x-1, l)$ . (13.14)

El valor del umbral en (13.12), sigue entonces de (13.13) y (13.14). □

Al tener una expresión explícita para los umbrales, conociendo  $\rho$ ,  $p$ , y la distribución de las ofertas, la expresión anterior puede ser calculada fuera de línea y usada para realizar subastas utilizando la política (13.9).

**Observación 13.3.** *En el caso del modelo simplificado analizado en la sección anterior, la optimización con horizonte reducido 13.9, se formula como:*

$$V_2(x^0) = \max_{a \leq C-x^0} \{ \bar{U}(a) + \rho E_{x^1} \bar{U}(C-x^1) \}. \quad (13.15)$$

*De esta ecuación se puede ver que el segundo sumando de la optimización no se modifica. Esto significa que el costo marginal es el mismo en ambos casos, pero en el modelo simplificado dicho costo se compara con una utilidad marginal que corresponde a los valores esperados de los estadísticos de orden de las ofertas. Por lo tanto, lo que se obtiene en este caso es una condición fija que no depende de los valores actuales de las ofertas y que indica cuántos usuarios dejar ingresar dependiendo sólo de la cantidad de circuitos ocupados antes de la subasta.*

**Ejemplo 13.1.** *Se evaluarán los resultados anteriores en algunos ejemplos simples (para  $\rho = 1$ ).*

*Para  $C = 1$ , hay un solo costo del enlace  $\bar{w}_1 = pE(b^{(1)})$ , que actúa como umbral de admisión para las ofertas recibidas cuando el circuito se libera. Para el caso de  $N$  ofertas, uniformemente distribuidas en  $[0, b_{max}]$  se tiene que  $\bar{w}_1 = p \frac{N}{N+1} b_{max}$ .*

*En este ejemplo si se adoptara el modelo simplificado habría que comparar la relación entre la utilidad marginal  $E(b^{(1)})$  y el costo marginal  $pE(b^{(1)})$ . Como se ve el costo marginal es siempre menor que la utilidad marginal (salvo en el caso límite con  $p = 1$ ). Por lo tanto, siempre que el circuito está libre se debe ocupar el enlace, lo cual es evidente porque la otra política estacionaria posible que sería no ocupar si el circuito está vacío, tiene ganancia nula (en el caso límite  $p = 1$  el circuito nunca se libera y también gana cero). Para el modelo simplificado, la política óptima con horizonte infinito y con horizonte reducido es la misma (en este caso con  $C = 1$  y  $\rho = 1$ ). Esto se*

puede ver fácilmente porque en este caso sólo hay una política estacionaria donde hay ganancia no nula. Como se verá en el último ejemplo, esta afirmación no es cierta para el modelo no simplificado, ya que hay una diferencia entre lo que se gana con horizonte infinito y con horizonte reducido.

Si  $C = 2$ , hay dos costos marginales:  $\bar{w}_1 = pE(b^{(2)})$  para ocupar la primera conexión, y  $\bar{w}_2 = p(E(b^{(2)})(1 - p) + E(b^{(1)})p)$  para ocupar la segunda. Para ofertas con distribución uniforme en  $[0, b_{max}]$  se tiene

$$\bar{w}_1 = p \frac{N - 1}{N + 1} b_{max}$$

$$\bar{w}_2 = p \frac{(N - 1)(1 - p) + Np}{N + 1} b_{max}.$$

Por último se compara la política de reducción del horizonte con la política óptima MDP con horizonte infinito en el caso de un circuito ( $C = 1$ ). En este caso la política óptima con horizonte infinito es también una política de umbral, pero el umbral óptimo no tiene una expresión analítica simple. Sin embargo, puede ser calculado numéricamente utilizando el algoritmo de value iteration utilizando el software [32]. En la figura 13.15 se muestran los umbrales de aceptación para las dos políticas: se puede ver que la política de horizonte infinito es más demandante. En la figura 13.16 se muestra la ganancia media obtenida por simulación para ambas políticas. Los resultados son muy similares. Por lo tanto, en este caso se ha logrado obtener casi la ganancia óptima mirando solamente un paso hacia el futuro. Por otro lado, si se aplica la política miope que siempre llena el enlace, la gráfica muestra que hay una notoria pérdida de ganancia.

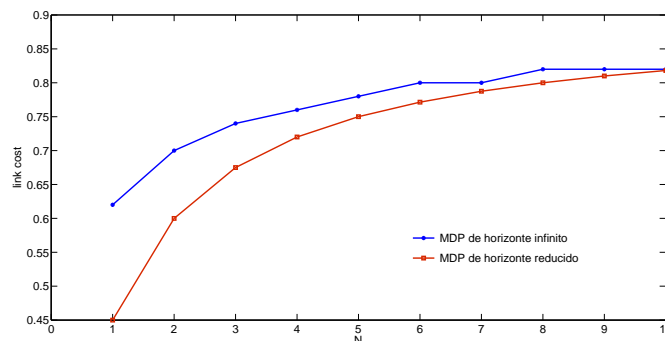


Figura 13.15: Comparación de umbrales entre políticas,  $C = 1$ ,  $p = 0,9$ .

## 13.7. Conclusiones

En primer término en este capítulo se propuso utilizar mecanismos de subastas para maximizar la ganancia a largo plazo del operador y controlar el acceso a una red con calidad de servicio. La formulación de este problema teniendo en cuenta el riesgo de que en el futuro lleguen ofertas mejores

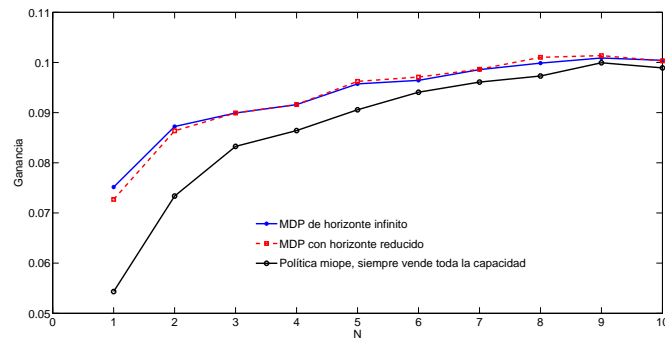


Figura 13.16: Comparación de la ganancia entre políticas,  $C = 1$ ,  $p = 0,9$ .

y deban ser rechazadas, es altamente más complejo que el problema habitualmente estudiando en el contexto de las redes de rematar toda la capacidad del enlace cada vez.

La formulación de este problema de subastas de la capacidad de un enlace como un Markov Decision Process es una propuesta original. Esta formulación permite entender además la complejidad del problema y los diferentes aspectos que influyen en su solución.

La propuesta de reducir el horizonte del MDP para poder encontrar una solución constituye también un aporte en varios sentidos. Por un lado, permite encontrar umbrales explícitos que pueden ser calculados fuera de línea y utilizados luego como reglas de decisión para controlar el acceso al enlace. Por otro lado, en los ejemplos vistos, aunque son simples para poder calcular la política óptima con el MDP de horizonte infinito numéricamente, se muestra que la ganancia obtenida por ambos MDPs es similar y sobre todo, significativamente mayor que la política miope. Además esta política de horizonte reducido permitirá abordar el problema de la red en el capítulo siguiente.

## Capítulo 14

# Asignación óptima de recursos en una red

### 14.1. Introducción

En este capítulo se estudia el problema de la optimización de la ganancia del operador de una red con topología general. En primer lugar se considerará un conjunto de usuarios que hacen ofertas por un cierto ancho de banda entre dos extremos de la red. La red deberá tomar la decisión respecto de cómo asignar la capacidad de sus enlaces con el objetivo de maximizar su ganancia. Se considerará en la sección 14.2 que se remata una única vez, es decir en cada subasta se buscará maximizar la ganancia si tener en cuenta el tiempo futuro. En esta primera parte se tendrá en cuenta sólo la topología de la red y se buscará el conjunto de requerimientos para tener una solución distribuida al problema de asignar los recursos.

En la sección 14.3 se analiza una aproximación fluida que permite introducir las consideraciones intertemporales en el caso de una red. En la sección 14.4, se extiende al caso más complejo donde intervienen consideraciones temporales al tener en cuenta el futuro en la optimización y se desarrolla un algoritmo distribuido para resolver este problema. En la sección 14.5 se analizan posibles implementaciones del algoritmo distribuido formulado en la sección 14.4 y se analizan diversos casos mediante simulaciones. En la sección 14.6 se explica porqué en este trabajo es más conveniente utilizar una subasta de primer precio que una tipo Vickrey. Por último, en la sección 14.7 se presentan las conclusiones de este capítulo.

### 14.2. El problema de optimización instantánea de los recursos de una red

#### 14.2.1. Notación y formulación del problema

Se recuerda la notación introducida en el capítulo anterior. La red estará compuesta de un conjunto de enlaces indexados por  $l$ , y un conjunto de rutas indexadas por  $r$ .  $R$  será la matriz de ruteo,  $R_{lr} = 1$  si la ruta  $r$  incluye al enlace  $l$ , en otro caso  $R_{lr} = 0$ .  $c = (c_l)$  es el vector de

las capacidades de los enlaces. Como se dijo antes, a cada ruta  $r$  está asociado un servicio con un requerimiento de ancho de banda  $\sigma_r$ . Pueden haber servicios diferentes con requerimientos de QoS diferentes sobre un mismo camino físico; en este caso se usará un índice diferente  $r$  para cada servicio diferente. Para cada ruta  $r$ , la red recibe un conjunto  $N_r$  de ofertas  $b_r^{(i)}$ , que son ordenadas de la siguiente forma:

$$b_r^{(1)} \geq b_r^{(2)} \geq \dots \geq b_r^{(N_r)}.$$

Se considerará también en este capítulo una subasta de tipo primer precio. Se define la variable  $\xi_{r,i}$  con  $\xi_{r,i} = 1$  si la oferta  $b_r^{(i)}$  es aceptada, y  $\xi_{r,i} = 0$  en otro caso. El problema de optimización de la ganancia del operador corresponde al siguiente problema de optimización entera:

$$\text{máx} \sum_r \sum_{i=1}^{N_r} b_r^{(i)} \xi_{r,i} \quad (14.1a)$$

$$\text{sujeto a} \sum_r \sum_{i=1}^{N_r} R_{rl} \sigma_r \xi_{r,i} \leq c_l \quad \forall l, \quad (14.1b)$$

$$\xi_{r,i} \in \{0, 1\}. \quad (14.1c)$$

Se utilizará también una forma alternativa para la formulación de este problema. Se observa que para una ruta  $r$  fija, todas las ofertas  $b_r^{(i)}$  son por la misma cantidad de ancho de banda (se recuerda que la ruta estaba asociada a un servicio), la solución óptima involucrará entonces las ofertas mayores en cada ruta,

$$\sum_{i=1}^{N_r} b_r^{(i)} \xi_{r,i} = \sum_{i=1}^{m_r} b_r^{(i)},$$

donde la variable entera  $m_r$  es el número de ofertas aceptadas en cada ruta. Se llamará  $a_r$  a la tasa asignada en cada ruta  $r$ ,  $a_r = \sigma_r m_r$ . Se define entonces

$$U_{b_r}(a_r) := \sum_{i=1}^{a_r/\sigma_r} b_r^{(i)}. \quad (14.2)$$

Esta función es definida para valores discretos de  $a_r$  (los múltiplos de  $\sigma_r$ ). Es conveniente extender esta función a todos los  $a_r \in \mathbb{R}$ , mediante interpolación lineal. Esta función lineal a tramos, es creciente y cóncava en  $a_r$ , dado que las ofertas son decrecientes. Con esta notación, se puede reescribir (14.1) de la siguiente forma:

**Problema 14.1 (Asignación óptima instantánea).**

$$\text{máx} \sum_r U_{b_r}(a_r) \quad (14.3a)$$

$$\text{sujeto a} \sum_r R_{rl} a_r \leq c_l \quad \forall l, \quad (14.3b)$$

$$a_r/\sigma_r \in \mathbb{Z}. \quad (14.3c)$$

### 14.2.2. Relajación convexa y solución distribuida

Si se ignoran momentáneamente las restricciones enteras en (14.3c), la optimización (14.3a-14.3b) tiene la misma forma que el problema de *maximizar la utilidad de la red* de la literatura de control de congestión [79, 92, 136]. En este caso la utilidad a maximizar es la ganancia del operador. Utilizando la solución dual se pueden buscar soluciones descentralizadas a esta relajación convexa. A continuación se resume este método brevemente.

Sea  $\alpha = (\alpha_l)$  el vector de los multiplicadores de Lagrange (los precios) asociados con las restricciones (14.3b), y sean  $q_r = \sum_l R_{rl}\alpha_l$  los precios acumulados por ruta. Se notará  $y_l = \sum_r R_{rl}a_r$ ,  $[\cdot]^+ = \max\{\cdot, 0\}$  y sea  $\gamma_l > 0$ . Entonces, el óptimo de (14.3a-14.3b) se puede encontrar dinámicamente a través del algoritmo de proyección del gradiente

$$a_r := \arg \max_{a_r} [U_{b_r}(a_r) - q_r a_r], \quad (14.4a)$$

$$\alpha_l := [\alpha_l + \gamma_l (y_l - c_l)]^+. \quad (14.4b)$$

En esta situación, (14.4a) utiliza los precios actuales de la ruta para establecer una tasa en dicha ruta con un máximo “excedente” (la utilidad menos un costo lineal). (14.4b) compara la asignación propuesta con la capacidad del enlace y actualiza los precios (hacia arriba o hacia abajo). El punto de equilibrio de (14.4a-14.4b) es un punto de silla del Lagrangiano

$$L(a, \alpha) = \sum_r U_{b_r}(a_r) + \alpha^T (c - Ra)$$

que corresponde a un optimización en  $a$ . En control de congestión, las ecuaciones anteriores son interpretadas como una descripción en el *plano de datos*, en el cual las fuentes elásticas adaptan su tasa de paquetes y los enlaces generan los precios en base al grado de congestión instantánea en cada uno de ellos.

Este algoritmo se sabe que converge asintóticamente al óptimo para utilidades estrictamente cóncavas [136].

En el caso que se está analizando, se deben pensar las ecuaciones anteriores como una iteración en el *plano de control*. Esto se debe a que la decisión de quienes ingresan, se toma antes de que las conexiones comiencen a enviar datos. En este caso se ejecuta este algoritmo mediante intercambio de mensajes de un protocolo de control para definir el resultado de una subasta antes de la asignación de recursos. En la sección 14.5 se describe una posible implementación de este mecanismo.

Un aspecto a considerar es que  $U_{b_r}$  no es estrictamente cóncava, es lineal a tramos, cambiando de pendiente en los múltiplos de  $\sigma_r$ . Por lo tanto (14.4a) puede tener múltiples óptimos; se puede utilizar este grado de libertad para seleccionar una solución que satisfaga las restricciones enteras en (14.3b), pero no es obvio que el algoritmo converja con esta elección. Si siempre lo hiciera, se podría concluir que la relajación convexa es exacta. Desafortunadamente, este no es el caso. La solución al problema relajado da una cota superior de la ganancia del problema entero (por ser una relajación uno del otro). Si se toma la parte entera de la solución al problema relajado se tendrá una cota inferior para el óptimo entero.

En el siguiente lema se muestra que si se resuelve el problema relajado por cada ruta a lo sumo hay un único usuario que recibe una fracción de la capacidad requerida en la ruta (los demás

reciben la capacidad requerida). Esto sólo indica que si en cada ruta la capacidad que se subasta es grande, la ganancia del problema entero y la del problema relajado estarán próximas. En este caso la diferencia entre la cota inferior y la superior es a lo sumo en la suma de las menores ofertas por cada ruta. Por lo tanto, la diferencia entre ambas cotas, será un porcentaje pequeño de lo que se gana por la cantidad de conexiones asignadas en todas las rutas. Sin embargo, como se verá en los ejemplos posteriores al lema, las asignaciones óptimas de ambos problemas pueden diferir considerablemente. El lema permite a partir de la solución óptima del problema convexo relajado obtener fácilmente una cota superior (la del problema convexo) e inferior (tomando la parte entera de menor oferta en cada ruta de la solución convexa) de la ganancia óptima del problema entero.

**Proposición 14.1.** *Se considera la siguiente relajación del problema 14.1*

$$\text{máx} \sum_r \sum_{i=1}^{N_r} b_r^{(i)} \xi_{r,i} \quad (14.5)$$

$$\text{sujeto a} \sum_r \sum_{i=1}^{N_r} R_{rl} \sigma_r \xi_{r,i} \leq c_l \quad \forall l, \quad (14.6)$$

$$\xi_{r,i} \in [0, 1]. \quad (14.7)$$

*En la solución de este problema, por cada ruta  $r$  hay a lo sumo una única oferta  $b_r^{(i)}$  que tiene  $0 < \xi_{r,i} < 1$ .*

**Demostración 14.1.** *Si planteamos el Lagrangiano correspondiente al problema 14.5:*

$$\begin{aligned} \mathcal{L}(\xi, \alpha, \beta, \gamma) &= \sum_r \sum_i^{N_r} b_r^{(i)} \xi_{r,i} \quad (14.8) \\ &+ \sum_l \alpha_l (c_l - \sum_{r \subset l} \sum_i \sigma_r \xi_{r,i}) \\ &+ \sum_{i,r} \beta_{i,r} (1 - \xi_{r,i}) + \sum_{i,r} \gamma_{i,r} \xi_{r,i} \end{aligned}$$

donde  $\xi = (\xi_{i,r})$  y donde  $r \subset l$  significa las rutas que contienen al enlace  $l$ .

Imponiendo las condiciones de optimalidad, se llega a que se debe cumplir que:

$$\begin{aligned} b_r^{(i)} &= \sum_{l \in r} \alpha_l \sigma_r + \beta_{i,r} - \gamma_{i,r} \quad (14.9) \\ \beta_{i,r} (1 - \xi_{i,r}) &= 0 \\ \gamma_{i,r} \xi_{i,r} &= 0 \end{aligned}$$

donde  $l \in r$  refiere a todos los enlaces  $l$  por los que pasa la ruta  $r$

Por lo tanto si para algún  $(i_1, r_1)$  se verifica que  $0 < \xi_{i_1, r_1} < 1$  entonces  $\beta_{i_1, r_1} = 0$  y  $\gamma_{i_1, r_1} = 0$ .



De lo anterior se deduce que para ese  $\xi_{i_1, k_1}$  se cumple que:

$$b_{r_1}^{(i_1)} = \sigma_{r_1} \sum_{l \in r_1} \alpha_l$$

pero como todas las conexiones de una misma ruta tienen el mismo ancho de banda  $\sigma_r$ , esta ecuación se puede verificar sólo para una conexión  $i_1$  de la ruta. Para las otras conexiones de la ruta, aquellas que tengan  $b_{r_1}^{(i)} > b_{r_1}^{(i_1)}$  tendrán  $\xi_{i, r_1} = 1$  y aquellas que tengan  $b_{r_1}^{(i)} < b_{r_1}^{(i_1)}$  tendrán  $\xi_{i, r_1} = 0$

Veremos a continuación dos ejemplos que muestran porqué las asignaciones óptimas de cada problema pueden ser diferentes

**Ejemplo 14.1.** Se considera en el primer ejemplo el caso de un único enlace. Por ese enlace se accede a dos servicios. La capacidad de enlace es  $C = 1,5$  unidades.

El servicio 1 tiene un requerimiento de ancho de banda  $\sigma_1 = 1$  unidad. Por ese servicio se reciben dos ofertas, de valores:

- $b_1^{(1)} = 1$
- $b_1^{(2)} = 0,9$

El servicio 2 tiene un requerimiento de ancho de banda  $\sigma_2 = 0,75$  unidad. Por este servicio también se reciben dos ofertas:

- $b_2^{(1)} = 0,6$
- $b_2^{(2)} = 0,5$

La solución al problema relajado es dejar ingresar la oferta 1 del servicio 1 y asignarle la mitad del ancho de banda requerido a la oferta 2 del servicio 1 y no dejar ingresar ninguna oferta del servicio 1. Sin embargo, la solución al problema entero es totalmente diferente, consiste en dejar ingresar las dos ofertas del servicio 2 y ninguna del servicio 1.

**Ejemplo 14.2.** Se consideran cuatro enlaces con capacidades  $c_l = 2$ , y cinco caminos (cada uno con requerimiento de ancho de banda  $\sigma_r = 1$ ), y con la siguiente matriz de ruteo

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Las ofertas por una misma ruta son todas iguales, con la siguiente distribución entre rutas:  $b_1 = b_2 = b_3 = b_4 = 1$ , y  $1 < b_5 < \frac{4}{3}$ . Entonces, el programa convexo relajado (14.3a-14.3b) tiene solución  $a^* = (\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, 0)$ , con ganancia óptima  $U^* = \frac{8}{3}$ . Para ver esto, se observa primero que  $a^*$  satisface (14.3b) con igualdad si se consideran los precios  $\alpha_l^* = \frac{1}{3}$ ,  $l = 1, 2, 3, 4$ , con precios agregados en las

rutas  $q^* = (1, 1, 1, 1, \frac{4}{3})$ . Dado que  $b_5 < q_5^*$ , se debe tener que  $a_5 = 0$ , pero las restantes coordenadas están indeterminadas en  $[0, 1]$ . Por lo tanto el punto propuesto  $(a^*, \alpha^*)$  es un punto de silla, pero esto puede no pasar con las coordenadas enteras de  $a$ .

En efecto, el programa entero en este caso puede ser resultado observando que a lo sumo dos conexiones pueden estar activas en todas las rutas, y por lo tanto la mejor solución es darle estas a las mejores oferta en la ruta cinco,  $\tilde{a} = (0, 0, 0, 0, 2)$ . Esto brinda una ganancia entera óptima  $\tilde{U} = 2b_5 < \frac{8}{3}$ . La solución óptima entera no sólo no se obtiene redondeando la solución relajada, sino que involucra una solución totalmente diferente.

Los ejemplos anteriores muestran que el problema de maximizar la ganancia instantánea del operador no es un problema de programación entera simple, su relajación convexa no es exacta.

Dado que los problemas de programación entera son NP-hard, se tiene una fuerte indicación de la complejidad de este problema, no es simple de resolver aún pensando en una solución centralizada y en este caso donde no hay consideraciones temporales. En la práctica esto fuerza a aceptar asignaciones sub-óptimas. Como dijimos el lema anterior permite fácilmente obtener cotas del error en la ganancia óptima entera.

**Observación 14.1.** *Este problema tiene similitudes con el problema de asignación de recursos estudiado en [115], bajo condiciones opuestas: demanda de ingreso a la red fija, y se busca minimizar un costo convexo sujeto a restricciones enteras. En ese trabajo también, salvo casos especiales, el programa entero no acepta una solución convexa relajada.*

### 14.3. Una aproximación fluida al problema de horizonte reducido

Se buscará ahora analizar el problema de optimización de la ganancia media del operador en el tiempo para el caso de una red. Si se formula el problema de horizonte reducido para el caso de una red (13.9) el primer paso del algoritmo de value iteration, se transforma en el problema de asignación instantánea visto en la sección anterior y correspondiente a la política miope:

$$V_1(x_0, b) = \max \sum_r U_{b_r}(a_r) \quad (14.10a)$$

$$\text{sujeto a } R(a + x_0) \leq c, \quad (14.10b)$$

$$a_r / \sigma_r \in \mathbb{Z}. \quad (14.10c)$$

El estado ahora es el vector  $x_0$  de circuitos ocupados en cada una de las rutas y el vector de todas las ofertas recibidas  $b$  expresado a través de las funciones utilidad  $U_{b_r}$  recibidas en cada ruta.

El segundo paso del algoritmo, donde se agrega a la ganancia actual la ganancia esperada a futuro mirando un paso hacia adelante, se transforma en

$$V_2(x_0, b) = \max \left\{ \sum_r U_{b_r}(a_r) + \rho E[V_1(x_1, b')] \right\} \quad (14.11a)$$

$$\text{sujeto a } R(a + x_0) \leq c, \quad (14.11b)$$

$$a_r / \sigma_r \in \mathbb{Z}. \quad (14.11c)$$

El segundo término se obtiene para cada valor posible de  $a_r$  de resolver (14.10) con las  $\bar{U}_{b_r}$ , y tomando como  $x_0$  en dicha optimización cada posible vector  $x_1$  de circuitos ocupados en cada ruta. De este conjunto de optimizaciones, se obtiene un conjunto de valores  $V_1(x_1)$ . Finalmente para obtener el segundo término de la optimización anterior, hay que ponderar cada  $V_1(x_1)$  obtenido por la probabilidad del vector  $x_1$  correspondiente, donde cada componente de  $x_1$ ,  $x_{1,r}$  tiene distribución  $Bin(x_{0,r} + a_r, p_r)$ .

Como se puede ver, el problema de optimización en el caso de una red y teniendo en cuenta el futuro aunque sólo sea a través de la reducción de horizonte es aún altamente complejo y fuertemente acoplado entre las diferentes rutas. Por este motivo para poder obtener una solución en este caso es necesaria una aproximación que permita encontrar una solución distribuida a dicho problema.

En primer término, se verá la aproximación a utilizar para el caso de la red en el caso de un enlace analizado en el capítulo anterior. Esta aproximación (que se denominará aproximación fluida) será la que se generalizará en la sección siguiente para el caso de una red.

Como se vio, el cálculo estocástico que involucra obtener la expresión de los umbrales de aceptación en el caso de un enlace (13.12) aparece difícil de generalizar para el caso de una red.

A partir de la ecuación de optimización con horizonte reducido

$$\max_{a \leq C - x^0} U_b(a) - \rho \bar{W}(x^0 + a) + \rho \bar{U}(C). \quad (14.12)$$

se reemplaza  $\bar{W}(x)$  por algo más simple de calcular y generalizar. Se define

$$\phi(x) = W(E[I_x]) \quad \text{para } I_x \sim \text{Bin}(x, p).$$

dado que  $W(\cdot)$  es convexa, se subestima el costo,  $\phi(x) \leq \bar{W}(x)$ . Sin embargo, si  $C$  es grande la distribución binomial estará concentrada alrededor de su media y el error será pequeño. Como resultado de esta aproximación, se tiene la siguiente expresión simple:

$$\phi(x) = W(px) = \bar{U}(C) - \bar{U}(C - px). \quad (14.13)$$

Esta función continúa siendo lineal a tramos y convexa, pero es más simple de calcular. La aproximación fluida a la política óptima se expresa entonces como:

$$\max_{a \leq C - x^0} U_b(a) - \rho \phi(x^0 + a) + \rho \bar{U}(C). \quad (14.14)$$

Si se introduce la variable auxiliar  $z$  se puede reescribir esta ecuación como un problema de optimización convexa:

$$\begin{aligned} & \max U_b(a) + \rho \bar{U}(z), \\ & \text{sujeto a } x^0 + a \leq C, \quad p(x^0 + a) + z \leq C. \end{aligned} \quad (14.15)$$

En el óptimo, la restricción en  $z$  es una igualdad,  $z = C - p(x^0 + a)$ . Se hace notar que  $z$  (la asignación futura esperada) no es necesariamente entera. Por otro lado,  $a$  debería ser entera; una debilidad de esta versión fluida es que en (14.14) los puntos de corte de  $\phi(x)$  no necesariamente son enteros. Por lo tanto, su solución puede conducir a resultados no enteros que en la práctica deberán ser aproximados. Se discutirá esto en la siguiente sección.

## 14.4. Subastas periódicas en el caso de una red

En esta sección se considera el problema general de optimización de la ganancia del operador en el tiempo sobre una red con topología general. En este contexto, el problema estudiado en la sección 14.2 corresponde al de la política miope de rematar todo el ancho de banda. Se desea ahora tener en cuenta en la optimización la ganancia futura, generalizando los resultados del caso de un enlace en la sección 13.5. Dada la complejidad de este problema, sólo se generalizará la aproximación fluida vista en la sección anterior.

Se describirá la decisión de asignación de recursos en el tiempo  $k = 0$ , y de esta forma se simplificará la notación omitiendo utilizar los índices de tiempo en las ofertas y demás variables. Se definen los vectores columna  $x^0$ ,  $a$ , y  $z$ , cuyas coordenadas por ruta  $r$  se notarán respectivamente la tasa  $x_r^0$  de ocupación previa, la tasa asignada en la subasta actual  $a_r$ , y la tasa esperada  $z_r$  de asignación en la próxima subasta ( $t = T$ ). Se toma la definición (14.2) de la utilidad lineal a tramos  $U_{b_r}(a_r)$  basada en las ofertas actuales para la ruta  $r$ . Análogamente se define  $\bar{U}_r(z_r)$ , donde se reemplaza la oferta por su valor esperado. Ambas se pueden expresar en función de  $\sigma_r$ , el requerimiento de ancho de banda del servicio asociado con  $r$ .

Otra característica del servicio asociado a una ruta es su duración que se expresa a través de  $p_r$  la probabilidad de que una conexión activa en  $t = 0$  en la ruta  $r$  permanezca activa en  $t = T$ <sup>1</sup>. Se notará  $P = \text{diag}(p_r)$  a la correspondiente matriz diagonal, el valor esperado del vector de las tasas en  $t = T^-$  vendrá dado por  $P(a + x^0)$ .

**Problema 14.2 (El problema de asignación de recursos en una red).**

$$\text{máx} \sum_r U_{b_r}(a_r) + \rho \bar{U}_r(z_r), \quad (14.16a)$$

$$s.t. R(a + x^0) \leq c, RP(a + x^0) + Rz \leq c, \quad (14.16b)$$

$$a_r/\sigma_r \in \mathbb{Z}. \quad (14.16c)$$

Para analizar el problema de optimización anterior será de utilidad reescribirlo de una manera similar al caso de un enlace. Para un vector  $x$  de tasas de las fuentes de información, se define

$$\phi(x) := \text{máx}_{Rz \leq c} \sum_r \bar{U}_r(z_r) - \text{máx}_{Rz \leq c - RP x} \sum_r \bar{U}_r(z_r). \quad (14.17)$$

Esta función juega un rol similar a la definida en la ecuación (14.13). El primer término (constante) se notará  $\bar{U}^{\text{máx}}$ , el cual tiene el papel de  $\bar{U}(C)$  en (14.13).<sup>2</sup>

Se debe notar que (14.17) es equivalente a un problema de optimización convexa en  $z$ , con  $x$  apareciendo de forma lineal del lado derecho de las ecuaciones asociadas con las restricciones. Se prueba en ([25], Ejercicio 5.32) que  $\phi(x)$  es convexa. Aún más, dado que la función objetivo es lineal a tramos se puede probar que  $\phi(x)$  es lineal a tramos también.

<sup>1</sup>para el caso de que el tiempo de servicio sea exponencial,  $p_r = e^{-\mu_r T}$ .

<sup>2</sup>Este término no es esencial, solamente se agrega para que  $\phi \geq 0$  y  $\phi(0) = 0$ .

A través de la función  $\phi$  se puede eliminar la variable  $z$  del problema 14.2, reduciendo el problema a la siguiente optimización en  $a$ , junto con las restricciones enteras (14.16c):

$$\max_{R(a+x^0) \leq c} \left[ \sum_r U_r(a_r) - \rho\phi(a+x^0) + \rho\bar{U}^{\max} \right] \quad (14.18)$$

**Observación 14.2.** *La función  $\phi(x)$  no tiene la propiedad de ser separable en sus componentes  $x_r$  del vector  $x$ . Por este motivo, la optimización anterior no puede ser separada a través de rutas diferentes, este problema está inherentemente acoplado.*

**Observación 14.3.** *Se considera la situación donde las restricciones en (14.18) son inactivas en el óptimo. Esto significa que, con las ofertas actuales es ventajoso dejar capacidad libre para la próxima subasta, el término predictivo juega un papel no trivial en este problema. En este caso, el óptimo puede encontrarse comparando la utilidad marginal (las ofertas) con el costo marginal, que toma un número finito de valores debido a que  $\phi$  es lineal a tramos. Genéricamente, entonces, estos valores se cruzan en un punto, como en el caso escalar. Sin embargo, como se observó, en este caso el punto encontrado no necesariamente satisface las restricciones enteras en  $a$ .*

La observación anterior implica que la relajación convexa (14.16) no es exacta. De todas formas, se utilizará para obtener una política que en algunos casos será óptima pero en general sub-óptima del problema entero. Se considera el Lagrangiano  $L(a, z, \alpha, \beta)$  dado por

$$\begin{aligned} L &= \left( \sum_r U_r(a_r) + \rho\bar{U}_r(z_r) \right) + \alpha^T(c - R(a+x^0)) \\ &\quad + \beta^T(c - Rz - RP(a+x^0)) \\ &= \sum_r [U_r(a_r) - (q_r + p_r v_r) a_r] + [\rho\bar{U}_r(z_r) - v_r z_r] \\ &\quad + \alpha^T(c - Rx^0) + \beta^T(c - RPx^0) \end{aligned}$$

$\alpha$  y  $\beta$  son los vectores de los multiplicadores de Lagrange (los precios) para cada una de las dos restricciones, y se han definido los precios agregados por ruta

$$q = R^T \alpha, \quad v = R^T \beta.$$

Se hace notar que el problema de optimización convexa formulado es similar a los problemas encontrados en la literatura de control de congestión (el problema de maximización de utilidad formulado por Kelly [79]). En este caso se agrega a la restricción habitual de la capacidad en los enlaces, una restricción adicional que proviene de la restricción de capacidad pero sobre la tasa de asignación media futura.

Este problema por tanto, al igual que los problemas de control de congestión se puede resolver por el método dual estándar, a través del algoritmo de proyección del gradiente, el cual en este caso

toma la forma:

$$a_r := \arg \max_{a_r} [U_{b_r}(a_r) - (q_r + p_r v_r) a_r]; \quad (14.19a)$$

$$z_r := \arg \max_{z_r} [\rho \bar{U}_r(z_r) - v_r z_r]; \quad (14.19b)$$

$$\alpha := [\alpha + \gamma (R(a + x^0) - c)]^+; \quad (14.19c)$$

$$\beta := [\beta + \gamma (RP(a + x^0) + Rz - c)]^+. \quad (14.19d)$$

El algoritmo anterior es muy similar al ya explicado en 14.2, y puede ser implementado de forma distribuida en el plano de control. En este caso como es natural se agrega un nuevo precio (multiplicador de Lagrange) asociado a la nueva restricción de capacidad en la tasa de asignación media futura y una nueva variable asociada con dicha tasa de asignación media futura. Este precio y esta nueva variable, deben ser enviadas entre las fuentes y los enlaces. Sin embargo, el mecanismo de pasaje de mensajes es en esencia el mismo.

(14.19a) obtiene la tasa comparando las ofertas con el precio umbral  $q_r + p_r v_r$ ; (14.19b) involucra a las ofertas esperadas, y el precio  $v_r / \rho$ .

El cálculo del algoritmo anterior hereda algunas dificultades de 14.2: si se imponen restricciones enteras en  $a_r / \sigma_r$  puede no alcanzar el equilibrio, pero una asignación sub-óptima puede encontrarse redondeando  $a_r$  en dirección decreciente.

## 14.5. Implementación y simulaciones

La implementación del algoritmo anterior en una red real puede ser realizado introduciendo algunas variantes a los protocolos de red actuales. Por ejemplo, la reserva de tasas y la señalización de precios puede ser realizada con el protocolo RSVP como se describe brevemente en el siguiente párrafo.

En primer lugar, las ofertas de los usuarios finales son recibidas por lo agentes intermediarios encargados de llevar a cabo de manera distribuida la subasta. Cada agente intermediario está asociado a un servicio y un camino desde el nodo de acceso a la red y el servidor correspondiente. Las ofertas serán recibidas hasta el momento en que se realice la subasta.

La asignación de la subasta se realiza entonces siguiendo el mecanismo (14.19), que se ejecutará en los elementos de red. Las variables asociadas a las reservas de tasas de cada ruta ( $a_r, z_r$ ) son enviadas por los agentes intermediarios a través de un mensaje RSVP PATH; los precios son acumulados a lo largo del camino con el mensaje de retorno RSVP RESERVATION. Esta iteración en el plano de control continúa hasta que converja, definido cuando se alcance un umbral de tolerancia o alternativamente luego de un cierto número de iteraciones. Para garantizar la factibilidad de la asignación final  $a_r$ , la que puede estar comprometida por la necesidad de una solución entera, estos valores finales son enviados en una ronda final de mensajes RSVP.

El esquema de funcionamiento anterior se puede ver en la figura 14.1.

Un aspecto importante de la implementación es que la función de utilidad media puede no ser conocida por el agente intermediario. En este caso, se utilizó un método adaptativo que estima

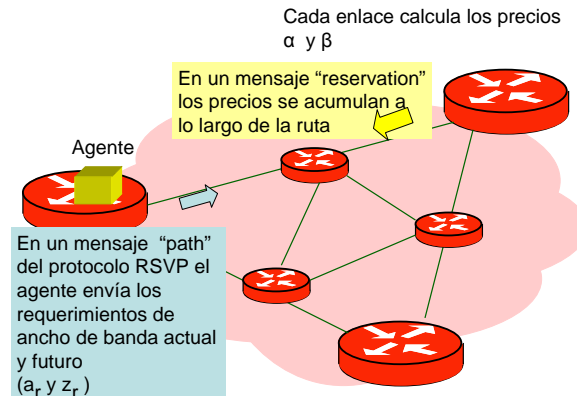


Figura 14.1: Implementación

la función  $\bar{U}$  a partir de las ofertas recibidas en las subastas anteriores, a través de un suavizado exponencial de las utilidades instantáneas. Esto es:

$$\bar{U}^{(k+1)}(z) = (1 - \alpha)\bar{U}^{(k)}(z) + \alpha U_{b^k}(z),$$

donde  $\bar{U}^{(k)}$  es la estimación actual. Se hace notar que este método sólo requiere de los valores de  $\bar{U}$  en los múltiplos de la tasa requerida para cada circuito. Aún más, esta iteración se aplica aún si el número de ofertas recibidas es aleatoria y varía en el tiempo.

Este proceso logra independizarse del mecanismo de asignación de la distribución de ofertas y también del proceso de arribo de las mismas. Si las ofertas arriban como un proceso aleatorio estacionario (por ejemplo Poisson),  $\bar{U}$  está bien definida, pero es difícil de escribir su forma explícita. Sin embargo, se puede estimar esta función a través del mecanismo explicado. En la figura 14.2 se muestra un ejemplo de estimación. En este caso, las ofertas arriban como un proceso de Poisson con intensidad  $\lambda = 10$  ofertas por subasta, y cada oferta es sorteada con una distribución uniforme en  $[0, 1]$ . El promedio es realizado sobre 100 períodos de subasta y con  $\alpha = 0,05$ . El valor real de  $\bar{U}(z)$  fue calculado numéricamente.

A los efectos de evaluar el algoritmo propuesto, se implementó en un simulador desarrollado en un proyecto de fin de carrera. Los estudiantes que desarrollaron el simulador fueron: De Andrés, Rattaro y Savergnini. El proyecto fue dirigido por A. Ferragut con mi co-tutoría. El simulador se implementó en Java. Es un simulador a escala de flujos que permite configurar una topología de red, con demandas de capacidad para cada servicio, distribución de ofertas y proceso de arribo de las mismas configurable. En el simulador también se implementó la política miope y el algoritmo de estimación de la utilidad referido antes. A continuación se presentan resultados para diferentes escenarios.

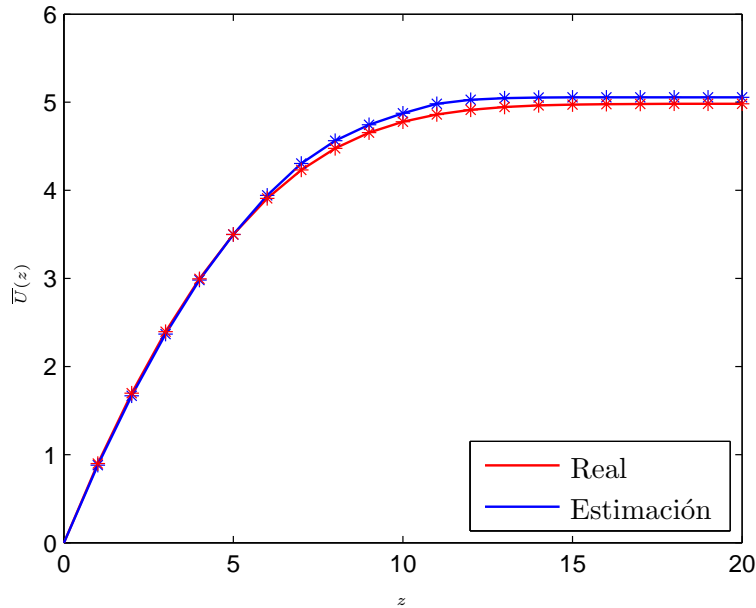


Figura 14.2: Estimación de  $\bar{U}(z)$  para un proceso de Poisson

#### 14.5.1. Escenario 1: subastas de la capacidad de un enlace.

Se compararán primero las políticas de reducción del horizonte y la política miope para el caso simple de un enlace con capacidad para 30 circuitos. Las subastas son realizados cada  $T$  minutos, y las ofertas arriban periódicamente con intensidad  $\lambda$  ofertas/min (que se asume fijo), totalizando  $N = \lambda T$  ofertas por subasta.

Las ofertas se sortean con distribución uniforme en  $[0, 1]$ , y las ofertas rechazadas son descartadas luego de cada subasta. Las ofertas aceptadas se asume que permanecerán en el sistema un tiempo con distribución exponencial de media 100 minutos. Por lo tanto,  $T$  es un parámetro crítico del sistema: aumentar  $T$  hace que más ofertas participen en cada subasta y también hace que más circuitos se liberen en cada tiempo entre subastas. Un valor muy grande de  $T$  disminuye la tasa de subastas, y por tanto disminuye la ganancia por unidad de tiempo.

En la figura 14.3 se muestra el resultado para  $\lambda = 0,5$ . En esta figura la política miope es comparada con la política de mirar un paso adelante en el futuro implementada con la distribución conocida de las ofertas y con el algoritmo de aprendizaje de la misma descrito antes. Se puede ver que en ambos casos la política de mirar un paso adelante tiene mayor ganancia por unidad de tiempo que la política miope.



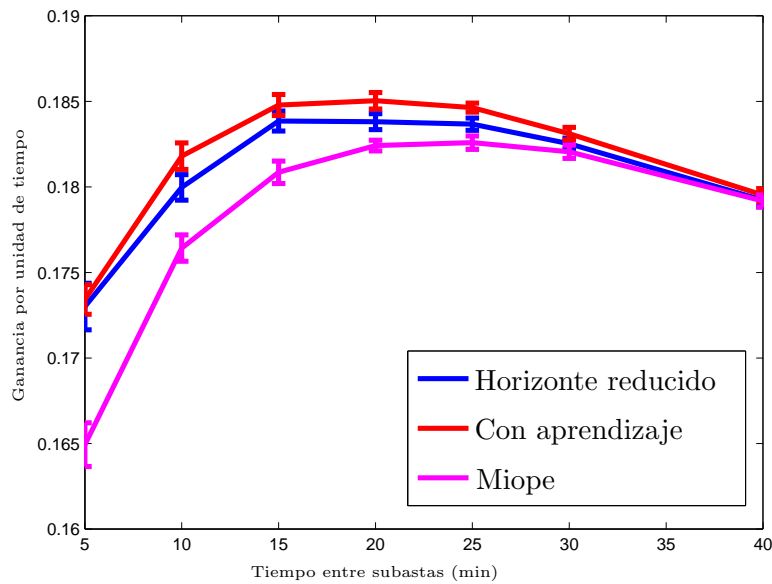


Figura 14.3: Un enlace: 30 circuitos, tasa de arribo de ofertas  $\lambda = 0,5$

### 14.5.2. Escenario 2: Red lineal

Se simulará ahora la topología de red que se muestra en la figura 14.4. En este caso, se espera que los usuarios de la ruta más larga 1 deban pagar más para que le sean asignados recursos dado que sus circuitos atraviesan dos enlaces. A los efectos de simular una situación más cercana a la realidad, la cantidad de ofertas por subasta no es fija sino que se distribuye como un proceso de Poisson con intensidad  $\lambda$  y se utiliza la política de horizonte reducido que mira un paso adelante. Se utiliza en este caso el algoritmo de aprendizaje de la distribución de ofertas.

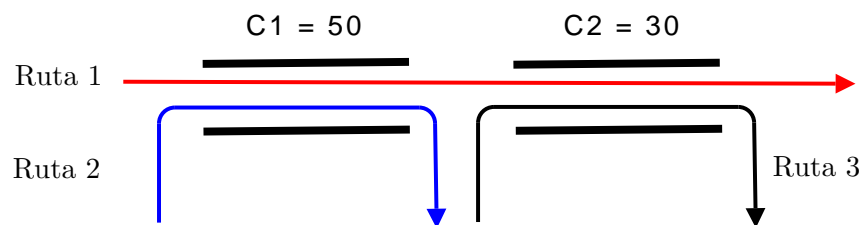


Figura 14.4: Red lineal con ofertas variables.

En la primera simulación se compara el resultado de esta política con la política miope, variando

la tasa de arribo de ofertas  $\lambda$  en los enlaces y manteniendo el tiempo entre subastas en  $T = 5$  min. Se fija la oferta media en la ruta 1 al doble de las ofertas medias en las rutas más cortas. Los resultados se muestran en la figura 14.5, donde se muestra la ganancia media por unidad de tiempo.

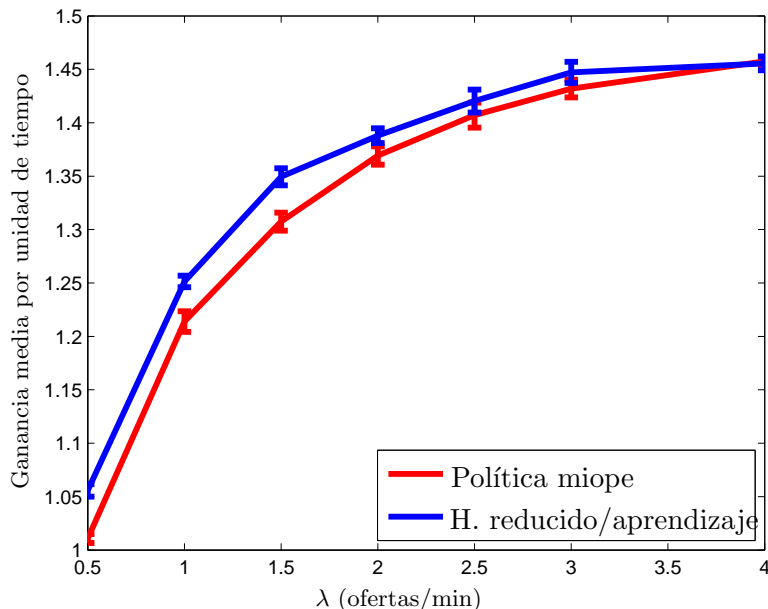


Figura 14.5: Red lineal variando la tasa de arribo de ofertas

Como se puede ver, también en este caso la política de mirar un paso hacia el futuro logra una ganancia significativa sobre la política miope para un amplio rango de tasas de arribo.

En el siguiente experimento, se varía la oferta media a lo largo de la ruta más larga. En este caso, se fija  $T = 5$  min. como antes y  $\lambda = 1$ . Se asumen ofertas independientes y uniformes de media 1 en las rutas cortas y se varía el valor medio de las ofertas en la ruta larga. Los resultados se muestran en la tabla 14.1. Como se puede ver, cuando la oferta media del agente intermediario 1 es el doble de la de los otros, se obtiene una asignación justa de los recursos. Si el agente 1 ofrece más, los recursos del agente 2 pasan a ser asignados al agente 1. Cuando la diferencia es mayor se obtiene que el agente 2 mantenga sólo 10 circuitos, y el agente 3 no logre obtener ningún recurso.

### 14.5.3. Escenario 3: Red overlay.

En este último escenario, se testea la factibilidad de la propuesta en un escenario más realista. Esta situación se muestra en la figura 14.6. En este caso se tienen cuatro servidores interconectados y varios agentes, cada uno de ellos busca obtener rutas a lo largo de la red overlay. Se tienen dos tipos de demandas: cada conexión sobre las rutas cortas 1 y 3 consume 2 circuitos representando al tráfico premium, y el resto consume 1 circuito. El número sobre los enlaces en la figura 14.6 indica

Cuadro 14.1: Efecto en la asignación de recursos de variar la oferta media.

Prom. oferta 1	$R_1$	$R_2$	$R_3$	$a_1$	$a_2$	$a_3$
0.5	0.003	0.643	0.428	0.3	47.5	28.4
1.0	0.051	0.627	0.401	2.8	45.0	25.9
1.5	0.252	0.558	0.312	9.5	38.4	19.2
2.0	0.583	0.469	0.191	17.3	30.7	11.5
2.5	0.918	0.397	0.115	22.1	25.9	6.7

- $R_r$ : ganancia por unidad de tiempo generado por la ruta  $r$ .
- $a_r$ : asignación media de recursos en la ruta  $r$ .

el número de circuitos disponibles. Se asume que la demanda de conexiones premium es menos frecuente (20%) pero las ofertas medias son el doble de las ofertas medias del tráfico no premium.

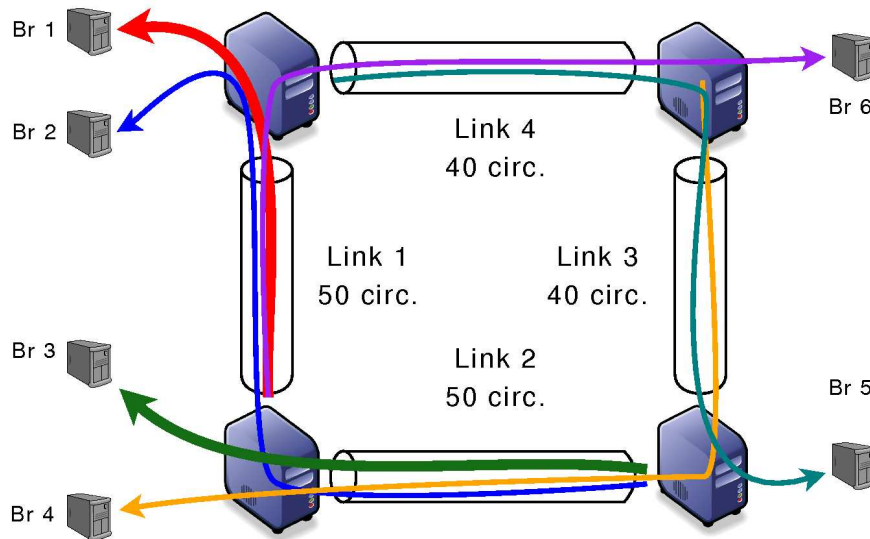


Figura 14.6: Ejemplo de red overlay.

Los resultados se muestran en la tabla 14.2. Como se puede ver, los usuarios premium que solamente usan un enlace obtienen un parte sustancial de los recursos.

Cuadro 14.2: Resultados de las simulaciones en el escenario 3

Agente	Br 1	Br 2	Br 3	Br 4	Br 5	Br 6
Enlaces	1	1-2	2	2-3	3-4	4-1
$R_r$	0.111	0.081	0.115	0.204	0.420	0.211
$a_r$	30.8	4.5	30.6	11.9	25.3	11.8

- $R_r$ : ganancia por unidad de tiempo generado en la ruta  $r$ .
- $a_r$ : tasa media asignada a la ruta  $r$ .

## 14.6. Consideraciones sobre la estrategia de los usuarios y la teoría de juegos

Buena parte de la literatura sobre subastas en internet se ha focalizado en el diseño de mecanismos que lleven al usuario a revelar su verdadera utilidad y que esta sea la estrategia dominante para los usuarios que participan en la subasta.

Las subastas tipo Vickrey o de segundo precio [144], donde al usuario que gana se le cobra por la segunda oferta más alta, son subastas donde la estrategia dominante es que el usuario revele su verdadera utilidad (esa es la estrategia que más le conviene).

En general, los mecanismos tipo VCG (por Vickrey-Clarke-Groves, ver por ejemplo [41]) tienen esta propiedad (denominada en inglés “incentive compatibility”). El hecho de lograr que los usuarios revelen su utilidad y que no les convenga “mentir”, es una condición muy importante para el diseño de muchos de los mecanismos propuestos para subastas en redes [87, 97, 42]. Por el contrario, en este trabajo se propone un mecanismo de primer precio, ignorando los aspectos estratégicos anteriores.

La pregunta que se procura analizar en esta sección es si con el mecanismo propuesto se está perdiendo algo importante para el objetivo del trabajo. Se buscará explicar en este análisis, que el mecanismo propuesto teniendo en cuenta los objetivos y las aplicaciones de este trabajo no sólo es un mecanismo adecuado, sino que además no sería preferible sustituirlo por un mecanismo tipo VCG.

La primera consideración es que los mecanismos tipo VCG, están motivados por políticas del “bienestar económico social”: asignarle recursos al subconjunto de usuarios tal que se alcance la máxima utilidad del conjunto. Esto no necesariamente genera la mayor ganancia para quien vende: aun más, los mecanismos que revelan la verdadera utilidad de los usuarios están subsidiados por el vendedor posiblemente a costa de su margen. Un resultado fundamental de la teoría de subastas, *El teorema de ganancia equivalente*, [41], establece que bajo ciertas hipótesis (principalmente, la neutralidad al riesgo de los participantes) todas las subastas tienen la misma ganancia esperada para los vendedores.

Sin embargo, bajo otras condiciones (compradores adversos al riesgo) las subastas de primer precio mejoran la ganancia de los vendedores [41].

Para comprender mejor los puntos anteriores, se considera el siguiente ejemplo simple. Se considera la subasta de la capacidad de un enlace de capacidad  $C$ , y se asume que en una subasta hay menos de  $C$  competidores (cada uno requiere una unidad de capacidad). En una subasta tipo

VCG se le cobrará a los ganadores el precio del primer oferente que quedó fuera, en este caso dicho precio es cero ya que nadie queda fuera, y por lo tanto la red no recibe ninguna ganancia. Si por el contrario se le cobra a los usuarios lo que ofrecen ¿cuál será el comportamiento de los compradores estratégicos? *Si ellos supieran* que la capacidad en esta subasta no es un recurso escaso, el comportamiento racional sería enviar una oferta cercana a cero. Esto confirma el teorema de ganancia equivalente. Sin embargo, en la práctica los usuarios no cuentan con esta información, por lo tanto no hay una respuesta clara a cuál sería el comportamiento racional, pero la oferta será de algún valor entre cero y su utilidad real. Si el usuario desea tener una buena probabilidad de asegurarse el circuito, muy posiblemente ofrezca una cantidad no despreciable. Por lo tanto, el vendedor obtendrá una mayor ganancia con una subasta de tipo primer precio.

Una segunda consideración también señalada en el trabajo de Maillé et al. [98], es la *complejidad* de los mecanismos que revelan la verdadera utilidad cuando se implementan en una red. Si se asume que se asigna capacidad como se discutió antes, y se acepta que el objetivo es maximizar el bienestar social, un mecanismo tipo VCG funcionaría aproximadamente de la siguiente forma en una red:

1. Para las ofertas actuales, se calcula la asignación óptima (máximo bienestar).
2. Para calcular cuánto cobrarle a un usuario: se remueve su oferta del sistema, y se recalcula la asignación óptima del sistema en esta nueva condición. La pérdida de bienestar que el usuario impone al conjunto por participar en la subasta es lo que se le cobra.

Ahora, cada una de estas optimizaciones necesarias para calcular cuánto cobrarle a cada usuario que ingresa al sistema es un problema de optimización equivalente al programa entero discutido en la sección anterior, que como hemos visto no es simple. No parece atractivo en cada subasta, resolver esos problemas de optimización una cantidad de veces del orden del número de usuarios. Por último, para poner a funcionar estos complejos mecanismos para responder a los jugadores estratégicos implica que se piensa que ellos son capaces de realizar estas complejas evaluaciones, ya que en otro caso no tendrían la información necesaria para “jugar” en este sistema. Es cuestionable que se pueda jugar este juego con la cantidad de información que se requiere para ser un jugador estratégico en él. Más aún si se agregan consideraciones intertemporales, que los usuarios deberían conocer la topología de toda la red, etc. Por lo tanto parece poco probable que sean aplicables este tipo de modelos de teoría de juegos en esta escala de problemas.

## 14.7. Conclusiones

En este capítulo se propone un mecanismo para asignar capacidad en una red a través de subastas periódicas. Dentro de las principales contribuciones de este capítulo se encuentra el haber desarrollado un modelo para este problema que puede ser resuelto mediante una aproximación fluida. Además, este modelo y su solución, resuelven mediante una optimización convexa el problema de maximizar la ganancia del operador teniendo en cuenta no sólo la ganancia actual sino también la futura, en una red con topología arbitraria y donde los recursos son asignados durante toda la duración del servicio. Otro aporte es encontrar para este tipo de problemas de subastas en red una forma de resolverlo similar a los algoritmos de control de congestión (“tipo Kelly”) lo que relaciona

dos campos hasta ahora no relacionados. Por último hay que resaltar, que mediante simulaciones se vio, que el algoritmo distribuido escala bien en diferentes topologías de red, que la estimación en línea de la distribución de probabilidad de las ofertas conduce a buenas aproximaciones y que el algoritmo mejora la política miope y el costo computacional para resolver uno u otro es equivalente.

## Parte IV

# Conclusiones de la tesis

## Capítulo 15

# Conclusiones

En este capítulo se quiere brevemente reseñar las conclusiones de la tesis y apuntar algunas líneas de trabajo futuro.

Como se mencionó en la introducción, el problema general que se analiza en la tesis es la posibilidad de obtener información de la calidad de servicio a lo largo de un camino en la red a los efectos de implementar mecanismos de control de admisión basados no sólo en la información de un enlace sino en la de todo el camino.

En el análisis de este problema se estudió la teoría de grandes desvíos y en particular la asintótica de muchas fuentes y buffer pequeño. Esta asintótica tiene la importancia de ofrecer un método para calcular las pérdidas en un camino. Se observó que este método era difícil de implementar en línea y se trabajó sobre la red ficticia. Se encontraron condiciones suficientes para que el análisis con la red ficticia coincida con el análisis en la red real. Además se encontraron cotas para el error cuando no coinciden y un algoritmo que permite fácilmente encontrar estas cotas. Además se observó en las simulaciones de diferentes escenarios que estas cotas eran buenas aproximaciones del valor real. Por último en esta área se trabajó en problemas de estimación de diversos parámetros de las funciones velocidad de grandes desvíos y se plantearon mecanismos de análisis y diseño de una red a partir de mediciones de tráfico. Con estos resultados, es posible mediante mediciones y utilizando el método de la red ficticia implementar algoritmos de control de admisión en línea. Sin embargo, también se observó que esta técnica tiene una debilidad importante para realizar estimaciones de extremo a extremo. Esa debilidad es que el camino para que pueda ser analizado por esta asintótica debe ser un camino de backbone y en general en muchos casos de interés existe uno o más enlaces de acceso a la red donde las hipótesis de esta asintótica no se verifican.

Para procurar solucionar el problema referido que plantea el análisis con la teoría de grandes desvíos se analizó en la siguiente parte de la tesis la estimación mediante mediciones activas de extremo a extremo. En esta área, por un lado se estudiaron las particularidades y se propusieron métodos de estimación de diferentes parámetros en el caso de redes con acceso a través de enlaces celulares. Por otro lado, se propusieron técnicas de estimación de la calidad de servicio que percibe una aplicación, utilizando diferentes herramientas de aprendizaje estadístico y mediciones activas con paquetes de prueba. Estas técnicas se probaron utilizando datos simulados con un simulador de redes dando muy buenos resultados. Se implementó un software que permite hacer mediciones activas en una red y que también implementa las técnicas de estimación propuestas. Con este



software se hicieron pruebas en diferentes redes operativas públicas y se lograron aproximaciones razonables.

Por último, se analizó el problema de una red con calidad de servicio desde la óptica de la tarificación. En este caso, se propuso un mecanismo de tarificación basado en subastas. se estudió dicho problema con detalle, y se formuló el problema de maximizar la utilidad media futura de un operador como un Markov Decision Process. Se propusieron mecanismos subóptimos para solucionar este problema tanto en el caso de un enlace como en el caso de una red con topología arbitraria. Para este último caso se encontró un algoritmo distribuido que permite resolver la optimización. Este algoritmo además se puede implementar con pequeñas modificaciones a protocolos ya existentes. Por último, en las simulaciones realizadas se observó que con las soluciones subóptimas encontradas se obtienen buenas aproximaciones.

En cuanto al trabajo futuro sin duda hay muchos temas aún abiertos. En el área de regresión funcional hay varios problemas para estudiar. Es interesante analizar SVM para datos funcionales u otras técnicas de regresión funcional y compararlos con los resultados obtenidos con Nadaraya-Watson funcional. También es interesante analizar el problema de cuál es la mejor medida de distancia entre los datos funcionales, en particular cuando se utiliza como variable funcional el estimador de la distribución empírica de la cola. Por otro lado, en cuanto a la regresión en  $\mathbb{R}^n$ , un tema interesante es encontrar un buen algoritmo que minimice la cantidad de características necesarias para la regresión. La integración de las técnicas desarrolladas en la tesis con herramientas de PQoS, si bien se esbozó en la tesis es otra área interesante de trabajo futuro.

También, en cuanto a líneas de trabajo futuro, una posible línea de acción es aplicar las técnicas de estimación con paquetes de prueba junto los métodos de aprendizaje estadístico a otros problemas de redes como por ejemplo la detección de anomalías. La mayoría de las técnicas de detección de anomalías desarrolladas se basan en aplicar diferentes herramientas observando el tráfico en un enlace o en un conjunto de enlaces de la red. La aplicación de los métodos desarrollados en la tesis a este problema puede aportar resultados interesantes.

En el área de tarificación, quedan también diversos problemas abiertos. Por ejemplo, estudiar la dinámica cuando un usuario que pierde en una subasta vuelve a la próxima modificando su oferta. También, analizar si puede haber otras aproximaciones diferentes a la aproximación fluida que se utilizó en el caso de la red y que permitan al resolver el problema convexo relajado obtener el óptimo del problema entero. Otro problema no estudiado en la tesis, en referencia a los remates en la red, es analizar que sucede cuando se admiten en la red múltiples caminos entre los nodos y el o los servidores.

Por último, cabe señalar que como resultado de esta tesis, se realizaron diez publicaciones con diferentes co-autores. También se debe señalar que con algunos problemas derivados de la tesis se desarrollaron: proyectos de fin de grado, tesis de maestría, se formularon y ejecutaron varios proyectos de investigación y se desarrollaron algunos convenios con ANTEL, que permitieron aplicar el conocimiento generado a problemas concretos del operador.

# Bibliografía

- [1] Adler M., Bu T., Sitaraman R., Towsley D., *Tree layout for internal network characterizations in multicast networks.*, NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication, 189–204, 2001.
- [2] Almes G., Kalidindi S., Zekauskas M., *A One-way Delay Metric for IPPM*, RFC 2679, 1999.
- [3] Almes G., Kalidindi S., Zekauskas M., *A One-way Packet Loss Metric for IPPM*, RFC 2680, 1999.
- [4] Almes G., Kalidindi S., Zekauskas M., *A Round-trip Delay Metric for IPPM*, RFC 2681, 1999.
- [5] Altman E., *Applications of Markov Decision Processes in Communication Networks: a Survey*, in Markov Decision Processes, Models, Methods, Directions, and Open Problems, E. Feinberg and A. Shwartz (Editors) Kluwer, pp. 488-536, 2001.
- [6] Andersen D., Balakrishnan H., Kaashoek F., Morris R., *Resilient overlay networks*, Vol. 35, No. 5., pp. 131-145, 2001.
- [7] Aspirot L., *Tesis de maestría en ingeniería matemática: Regresión no paramétrica para datos funcionales no estacionarios*, 2008.
- [8] Aspirot L., Bertin K., Perera G., *Asymptotic normality of the nadaraya-watson estimator for non-stationary functional data and applications to telecommunications*, a ser publicado en Journal of Nonparametric Statistics, 2009.
- [9] Aspirot L., Belzarena P., Bermolen P., Perera G., Simon M., *Quality of Service parameters and link operation point estimation based on effective Bandwidth- extended version*. Elsevier: Journal on Performance Evaluation. Special Issue on Performance Modeling and Evaluation of Heterogeneous Networks Volume 59, Issues 2-3, February 2005.
- [10] Baccelli F., Bremaud P., *Elements of queueing theory - palm martingale calculus and stochastic recurrences*, Springer-Verlag, 1994.
- [11] Beerends J., Hekstra A. , Rix A., Hollier M., *Perceptual Evaluation of Speech Quality (PESQ), the new ITU standard for end-to-end speech quality assessment. Part II – Psychoacoustic model*, 1998.

- [12] Belzarena P., Aspirot L., Perera G., Bazzano B., *End-to-end quality of service prediction based on functional regression*, Third International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NET's), 2005.
- [13] Belzarena P., Aspirot L., Perera G., Bazzano B., *End-To-End Quality of Service Prediction Based On Functional Regression, extended version*, In the book: Mobility Management and Quality-of-Service for Heterogeneous Networks, Chapter 19, River Publishers, 2009.
- [14] Belzarena P., Ferragut A., Paganini F., *Auctions for resource allocation in overlay networks*, In: Network Control and Optimization, Lecture Notes in Computer Science, aceptado para ser publicado en 2009.
- [15] Belzarena P., Ferragut A., Paganini F., *Network bandwidth allocation via distributed auctions with time reservations*, IEEE INFOCOM 2009, Rio de Janeiro, Abril 2009.
- [16] Belzarena P., Bermolen P., Simon M., Casas P. *Virtual Paths Networks Fast Performance Analysis* Performance Modelling and Evaluation of Heterogeneous Networks, HET-NET 04, 2004, Ikley, UK. HET-NETs 04. Technical Proceedings. 2004.
- [17] Belzarena P., Bermolen P., Simon M., Casas P. *Virtual Path Networks Fast Performance Analysis, extended version*, In the book: Mobility Management and Quality-of-Service for Heterogeneous Networks, Capítulo 17, River Publishers, 2009.
- [18] Belzarena P., Pereira J., Negreira J., Pérez S., *End-to-End Measurements over GPRS-EDGE Networks*, IFIP/ACM Latin America Networking Conference, LANC 07, San José, Costa Rica, 2007.
- [19] Belzarena P., González-Barbone V., Larroca F., Casas P., *Metronet: software para medición de calidad de servicio en voz y video*, 2006, Congreso Iberoamericano de Telemática, CITA 06, México, 2006.
- [20] Bertsekas D., *Dynamic Programming : Deterministic and Stochastic Models*, Prentice-Hall, 1987.
- [21] Berwin A., Turlach, *Bandwidth selection in kernel density estimation*, A review, Discussion Paper 9317, Institut de Statistique, Voie du Roman Pays 34, B-1348 Louvain-la-Neuve, Belgium, 1993.
- [22] Bessler S., Reichl P., *A Network Provisioning Scheme based on Decentralized Bandwidth Auctions* , Telecommunications Planning: Innovations in Pricing, Network Design and Management, Publisher: Springer; pp. 89-100, 2005.
- [23] Bianchi G., Borgonovo F., Capone A. Petrioli C., *Endpoint admission control with delay variation measurements for QoS in IP networks*, ACM Computer Communications Review, vol 32, pp. 61-69, April 2002.
- [24] Bolot J. C., *End-to-end frame delay and loss behaviour in the Internet*, Proc. ACM SIGCOMM, 289-298, 1993. 2000.

- [25] Boyd S., Vandenberghe L., *Convex Optimization*, Cambridge University Press, 2004.
- [26] Breslau L., Jamin S., Shenker S., *Comments on the performance of measurement-based admission control algorithm*, Proceedings of Infocom 2000(Tel Aviv, Israel), pp. 1233-1242, IEEE, March 2000.
- [27] Buschiazzo D., Ferragut A., Vázquez A., Belzarena P., *Fast Overflow Probability Estimation Tool for MPLS Networks*, IFIP/ACM Latin America Networking Conference - LANC 2005, Cali, Colombia. Memorias dela XXXI Conferencia Latinoamericana de Informática, 2005.
- [28] Buschiazzo D., Ferragut A., Vázquez A., *ARCA*, <http://iie.fing.edu.uy/investigacion/grupos/artes/arca>, 2004.
- [29] Casellas R., *MPLS traffic engineering*, Ph.D. thesis, ENST, Paris, 2002.
- [30] Cáceres R., Duffield N. G., Horowitz, Towsley D., *Multicast-based inference of network internal loss characteristics.*, IEEE Transactions on Information Theory Vol. 45, 2462–2480, 1999.
- [31] Casas P., Guerra D., Irigaray I., *Perceived Quality of Service in VoIP and Video IP services*, Technical Report, ARTES research group, Universidad de la República, 2005.
- [32] Chad I., Cros M-J., Garcia F., Sabbadin R., *Markov Decision Process Toolbox for MATLAB*, <http://www.inra.fr/internet/Departements/MIA/T//MDPtoolbox/>
- [33] Chang C-S., *Stability, queue length and delay of deterministic and stochastic queueing networks*, IEEE Transactions on Automatic Control, 39:913–931, 1994.
- [34] Chang C.-C., Lin C.-J., *LIBSVM: a library for support vector machines*, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [35] Chimento P., Ishac J., *Defining Network Capacity*, RFC 5136, 2008.
- [36] Clark D., Lehr W., Bauer S., Faratin, Sami R., Wroclawski J., *The Growth of Internet Overlay Networks: Implications for Architecture, Industry Structure and Policy*. In The Proceedings of the 34th Telecommunications Policy Research Conference (TPRC 05), Arlington, VA, USA, 2005.
- [37] Claypool M., Tanner J., *The Effects of Jitter on the Perceptual Quality of Video*, Proc. ACM Multimedia, (2), 115-118, 1999.
- [38] Courcoubetis C., Weber R., *Buffer overflow asymptotics for a switch handling many traffic sources*, Journal of Applied Probability, 1996.
- [39] Courcoubetis, Siris, *Procedures and tools for analysis of network traffic measurements*, Elsevier Science, 2001.
- [40] Courcoubetis C., Dramitinos M., Stamoulis G.D., *An Auction Mechanism for Bandwidth Allocation Over Paths* , 17th International Teletraffic Congress (ITC), December 2001.

- [41] Courcoubetis C., Weber R., *Pricing Communication Networks Economics, Technology and Modelling*, Wiley, ISBN: 978-0-470-85130-2, 2003.
- [42] Courcoubetis C., Dramitinos M. Stamoulis G., *An Auction Mechanism for allocating the bandwidth of networks to their users*, Computer Networks, September 2007.
- [43] Dacuna-Castelle D., Duflo M., *Probabilités et statistiques tome 2*, Masson, Paris, 1993.
- [44] Dembo A., Zajic T., *Large deviations: From empirical mean and measure to partial sums proces*, Stochastic Process Applications 57, 191–224, 1995.
- [45] Dembo A., Zeitouni O., *Large deviations techniques and applications*, 1993.
- [46] Demichelis C., Chimento P., *IP Packet Delay Variation Metric for IPPM*, RFC 3393, 2002.
- [47] Dovrolis C., Ramanathan P., Moore D., *What do packet disperssion techniques measure?*, Infocom 01, 905–914, 2001.
- [48] Downey A., *Using pathchar to estimate internet link characteristics.*, Measurement and Modeling of Computer Systems, 222–223, 1999.
- [49] Dramitinos M., Stamoulis G., Courcoubetis C., *Auction-based resource reservation in 2.5/3G Networks*, Mobile Networks and Applications, 9,557-566, 2004.
- [50] Dramitinos M., Stamoulis G. Courcoubetis C., *Auction-based Resource Allocation in UMTS High Speed Downlink Packet Access*, Proceedings of the First EuroNGI Conference on Traffic Engineering (NGI 2005), Rome, Italy, April 2005.
- [51] Duan Z., Zhang Z.L., Hou Y.T., *Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning*, ICNP'02, IEEE, 2002.
- [52] Eun D. Y., Shroff N., *Network decomposition in the many-sources regime*, Advances in Applied Probability, 2002.
- [53] Fahmy, S., Minseok K., *Characterizing overlay multicast networks*, Proceedings. 11th IEEE International Conference on Volume , Issue, 4-7, Page(s): 61 - 70, 2003
- [54] Ferraty F., Vieu P., *Nonparametric models for functional data, with application in regression, time-series prediction and curve discrimination*, J. Nonparametr. Stat. **16**, no. 1-2, 111–125, 2004.
- [55] Ferraty F., Vieu P., *Nonparametric functional data analysis: Theory and practice*, Springer Series in Statistics, Springer, New York, 2006.
- [56] Ferraty F., Goia A., Vieu P., *Functional nonparametric model for time series: a fractal approach for dimension reduction*, Test **11**, no. 2, 317–344, 2002.
- [57] Ganesh A. J., O'Connell N., *The linear geodesic property is not generally preserved by a fifo queue*, Annals of Applied Probability, 8(1): 98 – 111, 1998.

- [58] Ganesh A. J., O'Connell N., *A large deviation principle with queueing applications*, Stochastic and Stochastic Reports, 2002.
- [59] Garroppo R., Giordano S., Oppedisano F., Procissi G., *A Receiver Side Approach For Real Time Monitoring of IP Performance Metrics*, Proc. of the EuroFGI Workshop on IP QoS and Traffic Control, 169-176, 2007.
- [60] Gelenbe E., *Random Neural Networks with Negative and Positive Signals and Product Form Solution*, Neural Computation, 1, 502-511, 1989.
- [61] Hall T.A., *Objective Speech Quality Measures for Internet Telephony*, Voice over IP (VoIP) Technology, Proc. of SPIE, 128-136, 2001.
- [62] Hande P., Zhang S., Chiang M., *Distributed rate allocation for inelastic flows*, IEEE/ACM Transactions on Networking, 2008.
- [63] International Telecommunication Union, *The E-Model, A Computational Model for Use in Transmission Planning*, rec. ITU-T G.107, 2005.
- [64] International Telecommunication Union, *Transmissions impairments due to speech processing*, rec. ITU-T G.113, 2002.
- [65] International Telecommunication Union, *Methods for subjective determination of transmission quality*, rec. ITU-P P.800, 1996.
- [66] International Telecommunication Union, *Methodology for the subjective assessment of the quality of television pictures*, rec. ITU-R BT.500-11, 2002.
- [67] International Telecommunication Union, *Subjective video quality assessment methods for multimedia applications*, rec. ITU-T P.910, 1999.
- [68] International Telecommunication Union, *Objective quality measurement of telephone-band (300-3400 Hz) speech codecs*, rec. ITU-T P.861, 1998.
- [69] International Telecommunication Union, *Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, rec. ITU-T P.862, 2001.
- [70] Itkonen J., Lempiäinen J., *Air Interface Capacity for GPRS/EDGE over GSM Traffic Load* Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. IEEE, 2002.
- [71] Jacobson V., *Pathchar, a tool to infer characteristics of internet path*, Presented at the mathematical sciences research institute, 1997.
- [72] Jain M., Dovrolis C., *Pathload: A measurement tool for end-to-end available bandwidth*, Proceedings of Passive and Active Measurements (PAM), 14-25, 2002.
- [73] Jain M., Dovrolis C., *Pathload: A measurement tool for end-to-end available bandwidth*, In Proceedings of Passive and Active Measurements (PAM) Workshop, 2002.

- [74] Jain M., Dovrolis C., *End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput*, IEEE/ACM Transactions in Networking, 2003.
- [75] Jannotti J., Gifford D., Johnson, Kirk L., Kaashoek, Frans M., O'Toole J., *Overcast: Reliable Multicasting with an Overlay Network*, Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI), 2000.
- [76] Karagiannis T., Molle M., Faloutsos M., Broido A., *A nonstationary poisson view of internet traffic*, INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004.
- [77] Karlsson G., *Providing quality for internet video services*, Proceedings of CNIT/IEEE ITWoDC 98, (Ischia, Italy), pp. 133-146, September 1998.
- [78] Kesidis G., Walrand J., Chang C.S., *Effective bandwidth for multiclass markov fluid and other ATM sources*, IEEE ACM Transaction on Networking, 1 (4), 424-428, 1993.
- [79] Kelly F., Maulloo A., Tan D., *Rate control in communication networks: shadow prices, proportional fairness and stability*, Journal of the Operational Research Society, 39:237-252, 1998.
- [80] Kelly F., *Notes on effective bandwidth*, Stochastic Networks: Theory and Applications, Oxford University Press, 1996.
- [81] Kelly T., *An ECN probe-based connection acceptance control*, ACM Computer Communication Review, vol. 31, pp 14-25, 2001.
- [82] Klemperer P., *Auctions: Theory and Practice*, The Toulouse Lectures in Economics, Nuffield College, Oxford University, Oxford OX1 1NF, England <http://www.paulklemperer.org>, Princeton University Press, 2004.
- [83] Koodlim R., Ravikanth R., *One-way Loss Pattern Sample Metrics*, RFC 3357, 2002.
- [84] Lai K., Baker M., *Nettimer: A tool for measuring bottleneck link bandwidth*, Proceedings of the USENIX Symposium on Internet Technologies and Systems, 2001.
- [85] Lanckriet G. R. G., Cristianini N., Barlett P., El Ghaoui L., Jordan M. I., *Learning the kernel matrix with semi-definite programming*, Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, Morgan Kauffman, 2002.
- [86] Lawrence E., Michailidis G., Nair V. N., *Inference of network delay distributions using the EM algorithm.*, Technical Report, University of Michigan, 2003.
- [87] Lazar A. A., Semret N., *Design and Analysis of the Progressive Second Price auction for Network Bandwidth Sharing*, Telecommunication Systems - special issue on Network Economics, 2000.
- [88] Lewis D. P., Jebara T., Noble W. S., *Nonstationary kernel combination*, Proceedings of the International Conference on Machine Learning, Pittsburgh, PA, June 25-29, 2006.

- [89] Li Z., Mohapatra P., *QRON: QoS-aware routing in overlay networks*, Selected Areas in Communications, IEEE Journal on, Vol. 22, No. 1., pp. 29-40, 2004.
- [90] Likhanov, Mazumdar, *Cell loss asymptotics for buffers fed with a large number of independent stationary sources*, Proc. of IEEE, INFOCOM 98, 1998.
- [91] Lo Presti F., Duffield N. G., Horowitz J., Towsley D., *Multicast-based inference of network-internal delay distributions*, ACM/IEEE Transactions on Networking, Vol 10, 761–775, 2002.
- [92] Low S.H., Lapsley D.E., *Optimization flow control, I: basic algorithm and convergence*, IEEE/ACM Transactions on Networking, vol.7, no.6, pp 861-874, December 1999.
- [93] Machiraju S., Veitch D., Baccelli F., Nucci A., Bolot J.C., *Theory and practice of cross-traffic estimation*, SIGMETRICS '05: Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer system, 2005.
- [94] MacKie-Mason J., *A smart market for resource reservation in a multiple quality of service information network*, Technical Report, University of Michigan, URL: <http://wwwpersonal.umich.edu/jmm/research.html>, 1997.
- [95] McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York,, 1992.
- [96] Maille P., Tuffin B., *An Auction-Based Pricing Scheme for Bandwidth Sharing with History-Dependent Utility Functions*, in Proceedings of the First International Workshop on Incentive Based Computing (IBC'05), IEEE CS Press, Compiègne, France, September 2005.
- [97] Maillé P., Tuffin B., *Pricing the internet with multibid auctions*, IEEE/ACM Trans. Netw. 14(5): 992-1004, 2006.
- [98] Maille P., Tuffin B., *Why VCG auctions can hardly be applied to the pricing of inter-domain and ad hoc networks*, In Proceedings of the 3rd EURO-NGI Conference on Next Generation Internet Networks, Design and Engineering for Heterogeneity (NGI 2007). Trondheim, Norway. May 21-23 2007.
- [99] Masry E., *Nonparametric regression estimation for dependent functional data: asymptotic normality*, Stochastic Process. Appl. **115**, no. 1, 155–177, 2005.
- [100] McCanne S., Floyd S., *ns network simulator*, 2008.
- [101] Más I., Karlsson G., *Probe-based admission control for a differentiated-services internet*, Elsevier Computer networks, vol. 51, pp. 3902-3918, September 2007.
- [102] Mohamed S., Rubino G., *A Study of Real-Time Packet Video Quality Using Random Neural Networks*, IEEE Trans. on Circuits and Systems for Video Technology, 12, 1071-1083, 2002.
- [103] Mohamed S., Rubino G., Varela M., *Performance evaluation of real-time speech through a packet network: a Random Neural Networks-based approach*, Performance Evaluation, 57, 141-162, 2004.



- [104] Mokhtar S., Bazaraa Hanif D., Sherali C., Shetty M., *Nonlinear programming theory and algorithms*, John Wiley & Sons Inc, 1993.
- [105] MSN TV website, <http://msntv.com>
- [106] Multimedia in Java Applications, the Java Media Framework website, <http://java.sun.com/products/java-media/jmf>
- [107] Nadaraya È. A., *On estimating regression*, Theory of Probability and its Applications **9**, no. 1, 141–142, 1961.
- [108] Nadaraya È. A., *Nonparametric estimation of probability densities and regression curves*, Mathematics and its Applications (Soviet Series), vol. 20, Kluwer Academic Publishers Group, Dordrecht, 1989.
- [109] Negreira J., Pereira J., Pérez S., Belzarena P., *MetroCel: Estimación de performance sobre enlaces GPRS-EDGE*, 2007. <http://iie.fing.edu.uy/~javierp/MetroCel>, <http://iie.fing.edu.uy/javierp/MetroCel>
- [110] Noirie L., Post G., *Cov(t)-based traffic and queuing modeling*, Proceedings of the Next Generation Internet Networks, 2008. NGI 2008, IEEE, pp. 276–283, 2008.
- [111] O’Connell N., *Large deviations for departures from a shared buffer*, Journal on Applied Probability, **34**: 753–766, 1997.
- [112] Oliveira P., *Nonparametric density and regression estimation functional data*, Prepublication 05-09, Departamento de Matemática, Universidade de Coimbra, 2005.
- [113] Ong C. S., Smola A. J., Williamson R. C., *Learning the kernel with hyperkernels*, Journal of Machine Learning Research, **6**, 1043-1071, 2005.
- [114] Ostermann J., *Video coding with H.264/AVC: Tools, Performance, and Complexity*, IEEE Circuits and Systems Magazine, 2004.
- [115] Ozdaglar A., Bertsekas D., *Routing and Wavelength Assignment in Optical Networks*, IEEE/ACM Transactions on Networking, Vol 11, No. 2, April 2003.
- [116] Ozturk O., Mazumdar R., Likhanov N., *Many sources asymptotics in networks with small buffers*, Queueing Systems (QUESTA) (Vol.46, NOs. 1-2, pp.129-147), 2004.
- [117] Padhye J., Firoiu V., Towsley D., Kurose J., *Modeling TCP throughput: A simple model and its empirical validation*, Proceedings of the ACM SIGCOMM ’98 conference on applications, technologies, architectures, and protocols for computer communication, 303–314, 1998.
- [118] Pandora Radio website, <http://www.pandora.com>
- [119] Parmer G., West R., Fry G., *Scalable Overlay Multicast Tree Construction for Media Streaming*, in Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA ’07), June 2007.

- [120] Pavlidis P., Weston, J., Cai J., Grundy W. N., *Gene functional classification from heterogeneous data*, Proceedings of the Fifth Annual International Conference on Computational Molecular Biology, pp. 242-248, 2001.
- [121] Perera G., Simon M., Pechiar J., *Effective bandwidth estimation and testing for markov sources*, Eighth IFIP workshop on performance modelling and evaluation of ATM and IP networks, 2000.
- [122] Perera G., *Irregular sets and central limit theorems*, Bernoulli, **8**, no. 5, 627–642, 2002.
- [123] Puterman M. L., *Markov Decision Processes*, Wiley, New Jersey, 2005.
- [124] Rabinovitch P., *Statistical estimation of effective bandwidth*, Master's thesis, University of Cambridge, 2000.
- [125] Rajendran R., Ganguly S., Izmailov R., Rubenstein D., *Performance Optimization of VoIP using an Overlay Network*, NEC America, September, 2005.
- [126] Reichl P., Bessler S., Stiller B., *Second-Chance Auctions for Multimedia Session Pricing*, MIPS, pp 306-318, 2003.
- [127] Reichl P., Wrzaczek S., *Equilibrium Market Prices for Multi-Period Auctions of Internet Resources*, MMB, pp 25-34, 2004.
- [128] Ribeiro V., Riedi R., Baraniuk R., Navratil J., Cotrell L., *PathChirp: Efficient available bandwidth estimation for network paths.*, Passive and Active Measurement Workshop, 2003.
- [129] Ricciato F., Vacirca F. , Svoboda P. , *Diagnosis of capacity bottlenecks via passive monitoring in 3G networks: An empirical analysis*, Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 51 , Issue 4, Pages 1205-1231, 2007.
- [130] Rubino G., Varela M., Bonnin J. M., *Controlling Multimedia QoS in the Future Home Network Using the PSQA Metric*, The Computer Journal, 2006.
- [131] Semret, Liao, Campbell, Lazar, *Market pricing of differentiated internet pricing*, Proceedings of the 7th International Workshop on Quality of Service, pp. 184-193, 1999.
- [132] Silverman B.W., *Density estimation for statistics and data analysis*, 1986.
- [133] Strauss J., Katabi D., Kaashoek F., *A measurement study of available bandwidth estimation tools*, Internet Measurement Workshop, Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement, 39–44, 2003.
- [134] Shakkottai S., Srikant R., *Many-sources delay asymptotics with applications to priority queues*, Queueing Systems Theory and Applications (QUESTA) **39**, 183–200, 2001.
- [135] Shu J., Varaiya P., *Pricing network services*, in Proc. IEEE INFOCOM, vol. 2, pp. 1221-1230, 2003.

- [136] Srikant R., *The Mathematics of Internet Congestion Control*, Birkhäuser, Boston, MA, 2004.
- [137] Subramanian L., Stoica I., Balakrishnan H., Katz R.H., *OverQoS: An Overlay based Architecture for Enhancing Internet QoS*, First Symposium on Networked Systems Design and Implementation (NSDI'04), March, 2004.
- [138] Sun Microsystems, *Sun Developer Network (SDN)*, <http://java.sun.com>.
- [139] Sun Microsystems, *The Java Tutorial. Lesson: Applets*, <http://java.sun.com/docs/books/tutorial/deployment/applet/index.html>.
- [140] Tao S., Guérin R., *On-line estimation of internet path performance: an application perspective*, IEEE, Infocom 2004, 2004.
- [141] Tran H.T., *On the Service Overlay Network Concept for End-to-End QoS Assurances*, Technical report, FTW – TR – 2005 – 024, 2005.
- [142] Tsang Y., Coates M., Nowak R., *Network delay tomography*, IEEE Transactions on Signal Processing. Vol. 51, 2125–2136, 2003.
- [143] Vapnik V., *The nature of statistical learning theory*, 1995.
- [144] Vickrey W., *Counterspeculation, auctions and competitive sealed tenders*, Journal of Finance, 16, 8-7, 1961.
- [145] VideoLAN project website, <http://www.videolan.org>
- [146] Vieira S., Liebeherr J., *Topology Design for Service Overlay Networks with Bandwidth Guarantees*.
- [147] Vieira S., Liebeherr J., *Dynamic Topology Configuration in Service Overlay Networks: A study of Reconfiguration Policies*
- [148] Virgin Radio Online website, <http://www.virginradio.co.uk>
- [149] Voran S., *Objective Estimation of Perceived Speech Quality – Part I: Development of the Measuring Normalizing Block Technique*, IEEE Trans. on Speech and Audio Processing, 7 (4), 1999.
- [150] Voran S., *The Development Of Objective Video Quality Measures That Emulate Human Perception*, IEEE GLOBECOM, 1776-1781, 1991.
- [151] VQEG, *Final report from the video quality experts group on the validation of objective models of video quality assessment*, <http://www.vqeg.org/>, 2000.
- [152] Wang Z., Lu L., Bovik A. C., *Video Quality Assessment on Structural Distortion Measurement*, Signal Processing: Image Communication, 19 (2), 121-132, 2004.

- [153] Wang Z., *Rate scalable foveated image and video communications*, PhD thesis, Dept. of ECE, The University of Texas at Austin, 2001.
- [154] Wang Z., Lu L., Bovik A.C., *Why is image quality assessment so difficult?*, Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, 2002.
- [155] Watson G. S., *Smooth regression analysis*, Sankhyā Ser. A **26**, 359–372, 1964.
- [156] Wijnants M., Cornelissen B., Lamotte W., *An overlay network providing application-aware multimedia services*, ACM International Conference Proceeding Series, Vol. 198, Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications, Pisa, Italy, ISBN:1-59593-505-3, 2006.
- [157] Wischik D., *The output of a switch or effective bandwidths for networks*, Queueing Systems, **32**, 1999.
- [158] Wischik D., *Sample path large deviations for queues with many inputs*, Annals of Applied Probability, 2000.
- [159] Wolsey L.A., *Integer Programming*, John Wiley and Sons Inc, ISBN 0-471-28366-5, 1998.
- [160] Yang W., *Enhanced Modified Bark Spectral Distortion (EMBSD): An Objective Speech Quality Measure Based on Audible Distortion and Cognition Model*, Dissertation, Temple University, Philadelphia, USA, 1999.
- [161] YouTube website, <http://www.youtube.com/>
- [162] Zhang Y., Paxson V., Shenker S., *The stationarity of internet path properties: Routing, loss, and throughput*, Technical report, ACIRI, 2000.
- [163] Zhang Y., Duffield N., *On the constancy of internet path properties*, IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (New York, NY, USA), ACM, pp. 197–211, 2001.
- [164] Zhu Y., Li B., Guo J., *Multicast with network coding in application-layer overlay networks*, Selected Areas in Communications, IEEE, Volume: 22, Issue: 1 On page(s): 107- 120, 2004.
- [165] *Spatial-Temporal Distortion Metrics for In-Service Quality Monitoring of Any Digital Video System*, SPIE International Symposium on Voice, Video, and Data Communications, 1999.
- [166] Ziegler T., Tran H.T., *A design framework towards the profitable operation of service overlay networks*