

INGENIERÍA DE TRÁFICO EN LÍNEA EN REDES MPLS
APLICANDO LA TEORÍA DE GRANDES DESVIACIONES

Por
Pablo Belzarena

ENVIADA EN CUMPLIMIENTO PARCIAL
DE LOS REQUERIMIENTOS PARA OBTENER EL GRADO DE
MAESTRÍA EN INGENIERÍA ELÉCTRICA
de la
UNIVERSIDAD DE LA REPÚBLICA
MONTEVIDEO, URUGUAY
2003

© Copyright by Pablo Belzarena, 2003

UNIVERSIDAD DE LA REPÚBLICA
DEPARTMENT OF
TELECOMUNICACIONES

Los abajo firmantes certifican haber leído el presente documento y recomiendan a la Facultad de Ingeniería la aceptación de la tesis titulada **“Ingeniería de Tráfico en línea en redes MPLS aplicando la teoría de Grandes Desviaciones”** de **Pablo Belzarena** en cumplimiento parcial de los requisitos necesarios para obtener el grado de **Maestría en Ingeniería Eléctrica**.

Dated: 2003

Tutores:

Maria Simon, Daniel Kofman

Tribunal:

UNIVERSIDAD DE LA REPÚBLICA

Date: **2003**

Autor: **Pablo Belzarena**

Título: **Ingeniería de Tráfico en línea en redes MPLS aplicando
la teoría de Grandes Desviaciones**

Departamento: **Telecomunicaciones**

Degree: **M.I.E.** Convocatoria: **Noviembre** Año: **2003**

Firma del autor

DERECHOS DE COPYRIGHT
PERMISOS DE TERCEROS

Tabla de Contenido

Tabla de Contenido	v
List of Figures	VIII
0.1. Agradecimientos	x
1. Resumen: Introducción, Objetivos, metodología y principales contribuciones	1
1.1. Introducción	1
1.2. Objetivos de la Tesis	5
1.3. Organización de la tesis	5
1.4. Metodología y principales resultados	6
1.4.1. Ingeniería de Tráfico en línea en MPLS asegurando QoS	7
1.4.2. Modelos para asegurar QoS de punta a punta en una red MPLS	8
1.4.3. Estimación de parámetros de QoS	15
1.5. Conclusiones	16
2. Calidad de Servicio en redes IP	18
2.1. Introducción	18
2.2. ¿Donde se generan pérdidas, retardos y jitter en Internet?	20
2.3. ¿Como solucionar los problemas de calidad de servicio y aprovechar los recursos de la red?	22
2.4. Una Arquitectura para ofrecer QoS en IP: DiffServ	23
2.5. Conclusiones	27
3. Ingeniería de Tráfico en MPLS	28
3.1. Introducción: Conceptos básicos de la arquitectura MPLS	28
3.2. Ingeniería de tráfico en MPLS	30
3.2.1. Ruteo Explícito	31
3.2.2. Reparto de carga	36
3.3. Conclusiones	38

4. Grandes desviaciones aplicadas al análisis de performance en la asintótica de "buffer grande"	39
4.1. Introducción	39
4.2. Ecuación de Lindley y Teorema de Loynes	41
4.3. Definiciones básicas de la Teoría de Grandes Desviaciones y Teorema de Crámer	42
4.3.1. Teorema de Cramér	44
4.3.2. Teorema de GARTNER -ELLIS	48
4.4. El régimen asintótico de buffer grande	51
4.4.1. El análisis de un enlace	51
4.4.2. Teorema de Mogulskii	54
4.4.3. Teorema de Dembo-Zajic	58
4.4.4. Teorema de Ganesh-O'Connell	59
4.4.5. Principio de Contracción	59
4.4.6. LDP para el tamaño de la cola de un enlace	60
4.4.7. Extensión de los resultados anteriores al análisis de una red	61
5. Grandes desviaciones aplicadas al análisis de performance en el régimen asintótico de muchas fuentes	64
5.1. Introducción	64
5.2. El régimen asintótico de muchas fuentes	64
5.2.1. Introducción	64
5.2.2. Ancho de Banda Efectivo	65
5.2.3. Probabilidad de pérdida en un enlace	67
5.2.4. De un enlace al análisis de una red, la propuesta de Wischik	74
5.3. Ajuste de la asintótica	76
6. Análisis de performance en una red MPLS con buffers pequeños	77
6.1. Introducción	77
6.2. Análisis de performance de una red alimentada por muchos flujos y buffer pequeño	78
6.3. Aplicación de los resultados anteriores al análisis de performance de una red MPLS	84
6.4. Otros parámetros de calidad de servicio en estas redes	85
6.5. ¿Es posible trabajar con la red 'ficticia' en lugar de la real ?	85
6.5.1. Simulaciones	87
6.5.2. El análisis de una red con dos nodos	96
6.5.3. El análisis de una red más general	98
6.6. Conclusiones	102
7. Estimación del ancho de Banda Efectivo, el punto de operación y los parámetros de Calidad de Servicio de un enlace	104
7.1. Introducción	104
7.2. Estimación del ancho de banda efectivo	105

7.2.1.	Estimadores no paramétricos	106
7.2.2.	Estimadores paramétricos	106
7.2.3.	Simulación de los estimadores paramétricos y no paramétricos	108
7.2.4.	Dificultades para la aplicación del modelo paramétrico markoviano a trazas reales	109
7.3.	Estimación del punto de operación de un enlace	118
7.4.	Estimación de los parámetros de calidad de servicio de un enlace	121
7.4.1.	Diseño de un enlace basado en la estimación del ancho de banda efectivo	123
8.	Una propuesta para realizar Ingeniería de Tráfico y garantizar Calidad de Servicio en MPLS	128
8.1.	Introducción	128
8.2.	La arquitectura de red MPLS propuesta para ofrecer QoS	131
8.3.	Mediciones de tráfico a realizar en el modelo propuesto	132
8.4.	Bases para calcular los parámetros de Calidad de Servicio de cada agregado en línea	133
8.4.1.	Cálculo de performance a través de la 'red ficticia'	135
8.4.2.	Aproximación en línea del punto de operación	136
8.5.	Capacidad remanente equivalente y Cantidad de fuentes remanentes	139
8.6.	Ingeniería de tráfico en línea	141
8.6.1.	Reparto de carga en línea	143
8.6.2.	Ruteo basado en restricciones	147
8.6.3.	Otras consideraciones adicionales	149
8.6.4.	Control distribuido vs centralizado, Protocolos e información que se requiere difundir	154
8.7.	Resultados numéricos	155
8.8.	Fortalezas y debilidades de la arquitectura propuesta, posibles mejoras	164
8.9.	Conclusiones	166
9.	Conclusiones y trabajo futuro	167
A.	Apéndice: Software desarrollado	170
A.1.	Introducción	170
A.2.	Principales funcionalidades del software	170
A.3.	Interfaz de usuario	171
A.4.	Clases y documentación del software	173
	Bibliography	174

Índice de figuras

2.1. Arquitectura de un Nodo exterior en DiffServ	24
2.2. Arquitectura de un Nodo interior en DiffServ	25
4.1. Régimen asintótico de buffer grande	40
4.2. Régimen asintótico de muchas fuentes	40
4.3. Transformada de Fenchel-Legendre	46
4.4. Puntos expuestos	49
4.5. Teorema 4.4	54
4.6. $\widehat{S}_n(t)$ para $n = 10$	56
4.7. Un tubo de radio ε	57
5.1. Regimen de muchas fuentes	65
6.1. Red con dos nodos	87
6.2. Red Intree	88
6.3. Topologías de Simulación	90
6.4. Error al calcular las pérdidas en el enlace 2 con la Topología B en lugar de la Topología A	90
6.5. Ancho de banda efectivo a la entrada y a la salida del enlace, para $N=1$	91
6.6. Ancho de banda efectivo a la entrada y a la salida del enlace, para $N=20$	93
6.7. $\gamma(s, t)$ para $N=1$	93
6.8. $\gamma(s, t)$ para $N=15$	94
6.9. $\gamma(s, t)$ para $N=29$	94
6.10. Calculo del s óptimo	95
6.11. Gama(s, t) para $N=29, t$ fijo, C fijo, s y B variables	95

6.12. Análisis con dos nodos	97
6.13. Análisis red general	99
7.1. Regimen de muchas fuentes	105
7.2. Ancho de banda efectivo para una fuente fluida markoviana	109
7.3. Ancho de banda efectivo teórico y estimado para una fuente fluida markoviana para $t=1$ fijo	110
7.4. Ancho de banda efectivo teórico y estimado para una fuente fluida markoviana para $t=50$ fijo	110
7.5. Traza de tráfico mpeg	113
7.6. EBW con estimador markoviano variando el umbral, t fijo =1	114
7.7. Histograma de la traza entre valores de (0 y 250 kb/s)	116
7.8. Histograma de la traza entre valores de 1700 y 2500 kb/s	117
7.9. Comparación del EBW estimado de la traza	117
7.10. Punto de operación estimado y región de confianza con nivel 95 %	120
7.11. γ_n estimado, γ teórico e intervalo de confianza de 95 %.	122
7.12. Capacidad estimada, capacidad teórica e intervalo de confianza de 95 %.	124
7.13. Tamaño de buffer estimado, teórico e intervalo de confianza de 95 %.	125
7.14. Variación de $-\gamma$ con el tamaño de buffer	126
7.15. Variación de $-\gamma$ con la capacidad del enlace	127
8.1. Solución de la ecuación de optimización en s	137
8.2. Arquitectura del backbone del caso de estudio	155
8.3. Topología de LSPs establecidos en el backbone	157
8.4. Cantidad de fuentes por cada LSP estimada fuera de línea	158
A.1. ventana de usuario del soft desarrollado	171
A.2. ventana de usuario del soft desarrollado	172

0.1. Agradecimientos

Quiero agradecer a María Simón y Daniel Kofman directores de la tesis, por el tiempo dedicado y las diversas orientaciones que me han dado a lo largo del desarrollo de misma.

Al grupo Artes (Laura Aspirot, Paola Bermolen, Andrés Ferragult, Gonzalo Perera y María Simón), con quienes trabajé en estos años en el modelado estadístico de redes, lo cual ha enriquecido mis conocimientos sobre el tema y este trabajo en diversos aspectos.

Quiero agradecer también a Ramón Casellas y Sergio Beker del ENST- Paris, quienes trabajaron conmigo durante la pasantía en dicha institución. Especialmente le agradezco a Ramón quien me introdujo en el tema de ingeniería de tráfico en MPLS y con quien discutí diversos tópicos de los abordados en esta tesis.

Capítulo 1

Resumen: Introducción, Objetivos, metodología y principales contribuciones

1.1. Introducción

Internet se ha transformado en los últimos años en una red de muy alta difusión en cuanto al número de usuarios conectados. Esto ha sido visto por parte de los operadores como una oportunidad de ofrecer nuevos servicios a dichos usuarios además del tradicional servicio de email, ftp y navegación Web. Algunos de estos servicios son por ejemplo servicios de telefonía, videoconferencia, televisión, radio, etc. Estos nuevos servicios presentan requerimientos diferentes en cuanto a volumen de tráfico, calidad de servicio y seguridad.

Internet no fue pensada por sus diseñadores originales para trabajar en este contexto de servicios sino en un contexto académico con intercambio de información del tipo emails, o ftp.

El paradigma en que se ha basado el envío de paquetes en una red IP (protocolo base de Internet) ha sido la denominada política 'best effort'. Best effort implica que el usuario envía paquetes y la red y esta hace su mejor esfuerzo para hacerlos llegar al destinatario, no asegurando ningún tipo de calidad del servicio (pérdidas, retardos, etc.). Con este principio no es posible ofrecer servicios con requerimientos fuertes de Calidad de Servicio(QoS) en

cuanto a pérdidas retardos o jitter como exigen por ejemplo los servicios de voz o video interactivo. Protocolos superiores a IP (como TCP) han procurado solucionar el problema de la pérdida de paquetes básicamente reenviando paquetes si estos no llegan a destino. Esto resuelve los problemas de la transferencia tradicional de datos, pero este tipo de protocolos no puede ser usado para la transferencia de servicios interactivos en línea, en los que no es posible esperar por una retransmisión.

La comunidad de Internet ha realizado esfuerzos diversos en los últimos años para romper el paradigma actual y aproximarse a la calidad de servicio brindada por Red Pública Telefónica (PSTN).

El problema que hoy se plantea es diseñar la nueva arquitectura, las políticas, las metodologías y las herramientas necesarias para desplegar una red multiservicio capaz de asegurar los requerimientos de QoS necesarios para cada uno de los servicios ofrecidos. Un problema adicional es que este cambio debe ser gradual porque de otro modo no sería aplicable, ya que los operadores deberían perder una inversión en tecnología aún no amortizada en muchos casos.

Muchos de los esfuerzos realizados para transformar IP en una red de servicio convergentes están aún en su fase experimental y no han logrado imponerse de forma masiva. En paralelo nuevas propuestas surgen frecuentemente, fruto de una fuerte investigación en esta área. Aspectos básicos sobre cómo asegurar calidad de servicio en Internet, cómo medirla o estimarla, qué protocolos o tecnologías usar para brindar estos servicios aún generan controversias. Eso abre las puertas a un campo donde hoy se encuentra un fuerte desarrollo académico y comercial.

Como mencionábamos, se han propuesto diferentes modelos para brindar QoS en redes IP. El primer modelo propuesto fue el de Servicios Integrados (IntServ) [7], el cual procuraba establecer para cada flujo reserva de los recursos necesarios a lo largo de la red, para asegurar la calidad de servicio requerida. Este modelo tiene problemas de escalabilidad (reserva de

recursos por flujo) y por lo tanto se lo ha dejado de ver como una solución posible en el corazón de Internet donde convergen millones de flujos. Recientemente ha cobrado fuerte desarrollo el modelo de Servicios Diferenciados (DiffServ)[5]. Este modelo busca solucionar los problemas de escalabilidad de IntServ agregando los flujos en clases y procurando dar calidad de servicio a cada clase según los requerimientos de la misma. DiffServ es un área de importante desarrollo actual en relación a Internet. Sin embargo, DiffServ no es suficiente en IP para poder asegurar QoS. Esto se debe a que con los protocolos actuales de ruteo IP se termina hiperagregando tráfico en ciertas zonas de la red aunque otras estén subutilizadas. Por este motivo la calidad de servicio aún para las clases de más alta prioridad de DiffServ se puede deteriorar. Como consecuencia, para poder asegurar QoS en IP es necesario realizar ingeniería de tráfico. Ingeniería de tráfico significa ser capaz de distribuir el tráfico que arriba a la red de manera eficiente dentro de la misma. Tradicionalmente la ingeniería de tráfico en IP se realizó usando el modelo IP sobre ATM. Esta arquitectura tiene diversos problemas. Los principales problemas están referidos a la gestión de dos redes (IP y ATM), a la escalabilidad de la red y a la performance en redes de alta velocidad por la adaptación de la capa IP a la capa ATM.

La Arquitectura MPLS (MultiProtocol Label Swiching)[39] es una nueva arquitectura que habilita a realizar Ingeniería de Tráfico en redes IP [3]. La característica principal de MPLS que habilita a realizar ingeniería de tráfico es la de ruteo explícito. El ruteo explícito permite establecer caminos (Label Switchwd Path, LSP) predefinidos para los paquetes. Esto se realiza desde los routers de la frontera de la red. MPLS retoma en este sentido las bases sobre las que se diseñó ATM, al establecer 'caminos virtuales' para los flujos agregados. Sin embargo MPLS se integra dentro de la tecnología IP, no requiriendo el despliegue, la operación y la gestión de una tecnología diferente como era el caso de IP over ATM.

Al realizar ingeniería de tráfico en MPLS se genera la posibilidad de usar esta arquitectura para asegurar Calidad de Servicio. Esto se debe a que en el ruteo IP actual, los

paquetes se envían por la ruta de menor número de saltos generando hiperagregación de tráfico en ciertas zonas de la red y zonas donde la red está subutilizada. Las funciones de ruteo explícito de MPLS permiten enviar los paquetes por una ruta preestablecida o que se obtenga la misma analizando la carga de la red. De esta forma se puede por ejemplo, enviar los agregados de flujo con fuertes requerimientos de calidad de servicio por rutas específicas descongestionadas.

Algunos de los principales tópicos de investigación y desarrollo en el área de Ingeniería de Tráfico en MPLS son: el ruteo basado en restricciones y el reparto de carga.

El ruteo basado en restricciones (Constraint based routing, CBR) busca caminos entre puntos de la red que satisfagan un conjunto de restricciones explícitas. Estas pueden ser por ejemplo, que las pérdidas sean menores que un cierto valor y/o que el retardo punta a punta sea menor que 100 ms y/o que exista un ancho de banda mínimo. Sin embargo se ha observado [45], que el ruteo basado en restricciones para casi cualquier problema real es un problema NP-completo. Por esta razón se han propuesto en fechas recientes múltiples algoritmos heurísticos sub-óptimos para realizar CBR [31].

El balanceo de carga (load balancing) plantea el problema de dividir el tráfico de un agregado de flujos entre diversos caminos basados en algún criterio de optimalidad de la red. En los últimos años se han propuesto diversos algoritmos para realizar reparto de carga fuera de línea entre diversos caminos (LSPs) en una red MPLS por ejemplo utilizando modelos basados en el ancho de banda efectivo [37]. También se han propuesto algoritmos para balancear carga en línea en una red MPLS [20].

Otro aspecto relevante en esta área, es la integración de MPLS y DiffServ. Se han propuesto dos modelos para desplegar DiffServ sobre una red MPLS [23]. El primer modelo E-LSP, plantea transportar dentro de cada LSP diferentes clases de servicio (diferentes clases de despacho de paquetes (scheduling) y de prioridad de descarte), utilizando los 3 bits experimentales del cabezal MPLS (Exp), para distinguir la clase a la que pertenece

el paquete. El segundo modelo L-LSP plantea que en cada LSP se transporte una única clase de scheduling, aunque si pueden ir marcados paquetes con prioridades de descarte diferentes. La integración de MPLS y DiffServ y la conveniencia de usar un modelo u otro es un tema aún en desarrollo.

1.2. Objetivos de la Tesis

La tesis tiene dos objetivos principales:

- Analizar el 'estado del arte' en cuanto modelos y propuestas para hacer Ingeniería de Tráfico sobre MPLS y su relación con la QoS en Internet.
- Proponer una arquitectura que permita realizar ingeniería de tráfico en línea y asegurar QoS de punta a punta en MPLS.

1.3. Organización de la tesis

En el capítulo 2 se resumen los problemas y las propuestas existentes en cuanto a QoS en Internet.

En el capítulo 3 se resumen las características generales de MPLS y algunas propuestas para realizar Ingeniería de Tráfico.

En los capítulos 4, 5 se analizan los principales resultados de la teoría de grandes desviaciones y su aplicación al análisis de redes.

En el capítulo 6, se estudia una propuesta que será la base que se utilizará en la tesis para modelar una red MPLS utilizando la teoría de grandes desviaciones.

En el capítulo 7, se estudia el problema de estimar en línea diferentes parámetros del tráfico y de la calidad de servicio de una red.

En el capítulo 8 se propone la arquitectura, las modificaciones a los protocolos y algoritmos que constituyen la propuesta de esta tesis para realizar ingeniería de tráfico en línea, se analizan fortalezas y debilidades de la misma y se presenta una aplicación numérica.

1.4. Metodología y principales resultados

La primera etapa de esta tesis fue el estudio de la arquitectura MPLS, diferentes propuestas para hacer ingeniería de tráfico y el problema de la calidad de servicio en IP.

Luego de realizar el análisis anterior se abordó el problema definido en el segundo objetivo.

La primer interrogante que surgió entonces fue ¿qué es lo que necesita un operador de internet (ISP de aquí en más por su sigla en inglés) en esta área?

El objetivo de un ISP al desplegar una arquitectura que habilite a realizar Ingeniería de Tráfico y ofrecer garantías de QoS es:

1. Aprovechar al máximo la inversión realizada en recursos
2. Tener una arquitectura simple de operar y gestionar. Este aspecto es también muy relevante y en muchas propuestas no se tiene en cuenta. Si se propone una arquitectura que optimice los recursos de un ISP y la QoS ofrecida pero que sea muy difícil de operar y gestionar no será una buena propuesta. Minimizar las intervenciones de los operadores y la facilidad de configuración de la red son puntos relevantes. Este punto se relaciona con un paradigma que ha sido uno de los éxitos de Internet. Procurar que los routers del corazón de la red sean lo más 'tontos' posibles. Esto asegura escalabilidad y simplicidad de la configuración y la operación.
3. Poder ofrecer un modelo de transporte que permita distinguir clases de servicio y que permita por lo tanto diferenciar económicamente los diferentes servicios ofrecidos.
4. Tener la seguridad que se respetan ciertas restricciones en los parámetros de QoS, por ejemplo en pérdidas o retardo. Este punto es importante para evaluar diversas propuestas existentes. Muchas propuestas se basan en optimizar la red para lograr por ejemplo que el retardo global de la red sea mínimo. El problema del ISP no es minimizar el retardo o las pérdidas sino estar seguro que estos cumplan las restricciones requeridas por cada clase. Es

más importante verificar que se cumplen las condiciones de QoS, que resolver un problema de optimización global de la red. Además se puede tener un óptimo global pero no cumplir los requerimientos de QoS de algún agregado de flujo.

Sobre estos cuatro puntos hemos procurado proponer mejoras a las arquitecturas existentes para realizar Ingeniería de Tráfico en MPLS y asegurar que se cumplan las restricciones de QoS.

1.4.1. Ingeniería de Tráfico en línea en MPLS asegurando QoS

La propuesta de esta tesis asume la existencia de un modelo híbrido de ingeniería de tráfico. Fuera de línea se define una política de ingeniería de tráfico y en línea se procura ajustar dicha política para poder cumplir con los requerimientos de QoS ante variaciones del tráfico y la red. La tesis se concentrará en la segunda parte de este modelo, es decir en la ingeniería de tráfico en línea.

El ruteo basado en restricciones es uno de los pilares de la ingeniería de tráfico en MPLS. Pero como mencionamos este es un problema complejo. Si bien existen muchas propuestas académicas, el algoritmo básico que se utiliza en la actualidad para hacer CBR es CSPF (Constrained Shortest Path First). La idea de este algoritmo es modificar el tradicional algoritmo SPF (Shortest Path First), podando previamente aquellos enlaces que no cumplen alguna restricción. En cuanto a restricciones asociadas a requerimientos de QoS en la práctica la única que se utiliza es la capacidad requerida por el camino a establecer respecto de la capacidad disponible del enlace. Esto se debe a que las condiciones que se pueden imponer como restricciones con este algoritmo son asociadas a métricas llamadas cóncavas (la restricción global la determina el mínimo de las métricas de cada enlace del camino). Esto lo verifica el ancho de banda (está determinado por el enlace de menor ancho de banda del camino). Sin embargo las pérdidas o el retardo son restricciones que responden a métricas en un caso multiplicativas (las pérdidas) y en otro caso aditivas (el retardo).

Este algoritmo no asegura pérdidas o retardo, salvo que se reserve el pico máximo de la fuente como ancho de banda requerido y en cuyo caso se genera una fuerte sub-utilización de los recursos.

Un objetivo de esta tesis es buscar una métrica cóncava y modificar este algoritmo de manera de poder asegurar pérdidas y/o retardos a diferentes clases de tráfico.

Establecer un nuevo LSP implica un cambio importante en la configuración de la red. Si es posible, es mejor re-enrutar el tráfico por caminos ya existentes. Por lo tanto la propuesta de la tesis de modificación del algoritmo CSPF, se integrará con una propuesta para balancear carga en línea entre diferentes caminos.

Una contribución de esta tesis ha sido definir una arquitectura que, con cambios razonables de implementar, modifique las propuestas existentes para lograr los objetivos planteados.

Cuando se comenzó a diseñar esta arquitectura surgieron dos aspectos fundamentales que se requerían para poder desarrollar la propuesta:

- Un modelo que permita evaluar performance y calcular los parámetros de QoS de extremo a extremo de la red.
- Una metodología para medir o estimar en línea parámetros que caracterizan el tráfico y la QoS de la red.

Resumiremos en las próximas dos secciones estos dos puntos que son la base para el desarrollo de la propuesta.

1.4.2. Modelos para asegurar QoS de punta a punta en una red MPLS

Al abordar este problema estudiamos las diferentes propuestas existentes sobre el mismo. En cuanto al análisis de performance de redes encontramos dos vertientes básicas. El análisis de performance por métodos estadísticos y el análisis por métodos determinísticos.

Las herramientas de modelado y análisis de performance de redes por métodos determinísticos se basan en los conceptos introducidos por Cruz [13],[14]. Esta teoría se conoce también como Network Calculus. A partir del trabajo de Cruz han aparecido diversos resultados para estudiar la performance de redes con estas herramientas [9],[10],[6]. Estos modelos conducen a resultados analíticos potentes pero presentan el inconveniente que trabajan sobre hipótesis de peor caso, lo que lleva a resultados muy conservadores. No se trabajará con estas herramientas a lo largo de la tesis.

Históricamente las redes de Telecomunicaciones fueron modeladas y analizadas utilizando herramientas de la Teoría de Colas clásica [29], [38],[16], [4], [30]. En esta teoría existen resultados analíticos de mucha utilidad para sistemas M/M/K, es decir para modelos donde el tiempo entre arribos y el de servicio son independientes y ambos con distribución exponencial. También existen resultados para modelos G/M o M/G, es decir donde el tiempo entre arribos ó el tiempo de servicio tiene distribución exponencial pero el otro tiene una distribución cualquiera. Estos últimos resultados son de todas formas más limitados que los obtenidos para el caso M/M/K. Sin embargo en casos más generales en que se tiene una cola a la que arriban fuentes de diferentes clases de tráfico con una distribución cualquiera y el servicio es otro proceso no exponencial, los resultados analíticos son más débiles, y de difícil aplicación.

Esta tesis requiere modelos que permitan la extensión del análisis una cola a una red de colas. En la teoría de colas clásica existen resultados para redes en condiciones particulares (como las redes de Jackson), pero las extensiones son muy limitadas para casos más generales.

Hace algunos años diversos investigadores comenzaron a utilizar otras herramientas estadísticas para abordar el problema de modelado y análisis de performance de redes de telecomunicaciones. Basados en que muchos de los fenómenos que interesan estudiar en un enlace de una red son lo que podríamos llamar eventos 'raros'. Cuando decimos 'raros' se

hace referencia a eventos como la pérdida de paquetes en un enlace de una red cuya probabilidad en una red actual puede tomar valores del orden de 10^{-6} o menores. Estos fenómenos están asociados a las colas de las distribuciones de interés (por ejemplo la distribución de la ocupación del buffer de un enlace), y estos eventos son grandes desvíos respecto de los valores medios de dichas distribuciones. Por este motivo ha cobrado mucha fuerza en el análisis de redes de telecomunicaciones la teoría de grandes desvíos. Esta teoría busca encontrar la velocidad con la que tiende a cero la probabilidad de un evento. Por ejemplo, la velocidad a la que tiende a cero la probabilidad de pérdida de un enlace cuando el tamaño del buffer tiende a infinito. La teoría de grandes desvíos presenta como ventaja que los resultados obtenidos son en general tratables analíticamente para colas de tipo muy generales. Sin embargo también presenta limitaciones en cuanto a la precisión de los resultados que se obtienen cuando se estima por ejemplo la probabilidad de pérdida a partir de sus resultados. La teoría de grandes desvíos dice cual es la asíntota a escala logarítmica con la que tiende a cero la probabilidad de pérdida. Una estimación de las pérdidas a partir de esta asíntota a escala logarítmica comete un error el cual puede en algunos casos llegar a ser importante.

El mayor desarrollo de estas herramientas aplicadas al estudio de un enlace de una red ha tenido lugar a partir de la introducción de la noción de ancho de banda efectivo por Kelly [28]. A partir de la noción de ancho de banda efectivo diversos autores han encontrado expresiones que permiten estimar la probabilidad de pérdida en un enlace de una red.

En la aplicación de grandes desvíos a las Telecomunicaciones se han estudiado dos tipos de regímenes: el régimen asintótico de buffer grande y el de muchas fuentes. En el régimen asintótico de 'buffer grande' se estudia el comportamiento de un enlace cuando el tamaño del buffer tiende a infinito. La asíntota de 'muchas fuentes' estudia el comportamiento cuando el número de fuentes tiende a infinito y se escalan proporcionalmente al número de fuentes, la capacidad y el tamaño del buffer del enlace.

En un backbone de internet (que es el interés principal en la tesis), es mucho más

adecuado el modelo de la asintótica de 'muchas fuentes' que el de 'buffer grande'. En el corazón de internet es más razonable asumir que arriban muchas fuentes a un enlace de la red, que asumir que el enlace tiene un tamaño de buffer muy grande. En un enrutador de acceso sin embargo puede ser más adecuado el modelo de 'buffer grande'.

En la tesis tendremos como objetivo analizar qué resultados existen en esta teoría para estudiar performance de redes y no sólo de un enlace.

Resultados para análisis de redes en el régimen de buffer grande

En el capítulo 4 analizamos las aplicaciones de grandes desvíos a redes de Telecomunicaciones en el régimen asintótico de buffer grande. Como se mencionó este régimen no es el más adecuado para modelar un backbone del corazón de Internet que es el interés principal de la tesis. Sin embargo por completitud del estudio de las aplicaciones de grandes desvíos y por sus aplicaciones a otro tipo de redes como las redes de acceso lo hemos incluido en la tesis. El objetivo de este capítulo es estudiar que resultados existen para analizar un enlace en este régimen y sobre todo, los resultados existentes para el análisis de una red que opere en este régimen.

En el régimen asintótico de buffer grande, hay resultados analíticos para el estudio de performance de un enlace como los que se muestran en el teorema 4.4.

Para extender estos resultados al análisis 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 [33] y se analiza en el teorema 4.9.

En principio este resultado permitiría aplicando los teoremas utilizados 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 la performance de cualquier enlace. Sin

embargo para que esto sea cierto el tráfico de salida debe verificar las mismas hipótesis que se le exigen al tráfico de entrada en el teorema 4.9.

En un trabajo reciente Ganesh y O'Connell [24] han encontrado que en general no es cierto que el tráfico de salida del enlace verifique todas las hipótesis del teorema 4.9. La hipótesis conocida como la propiedad de 'geodésica lineal', no es necesariamente verificada por el tráfico de salida. Esta hipótesis se refiere a que en el régimen asintótico de 'buffer grande', la trayectoria 'más probable' del gran desvío es lineal. Esto se deriva de que la función de velocidad puede ser expresada como la integral de una función convexa. En el trabajo de Ganesh y O'Connell se muestran contra ejemplos en los que esta propiedad no se preserva al pasar por un enlace. Primero se estudia un contraejemplo para el caso en que se tiene una sola clase de tráfico de entrada, y se ve que la salida no necesariamente cumple con la hipótesis. Si bien para este caso esta propiedad no se conserva en general, sí lo hace si el servicio del enlace es determinístico. Este último es el caso estudiado por Chang [9], para el análisis de un tipo particular de redes 'intree networks'. Para el caso en que se tienen diversas clases de tráfico a la entrada del enlace se muestra con un contraejemplo en el trabajo de O'Connell, que aún en el caso de servicio determinístico, la propiedad de geodésica lineal no necesariamente se preserva al atravesar un enlace. En el trabajo del Dembo y Zajic [17] se demuestra que si se cumplen hipótesis de mixing del tráfico de entrada, entonces se satisface la hipótesis de 'geodésica lineal'. Lo que sucede es que al atravesar un enlace con un buffer grande se puede introducir dependencia 'fuerte' en el tráfico y eso puede llevar a que se deje de cumplir esta hipótesis. La conclusión respecto al interés de esta tesis es que en este régimen, para el tipo de redes que interesan (multi-servicio y feed forward con una topología general), no se tienen aún resultados suficientemente potentes que permitan estudiar performance de punta a punta.

Resultados para análisis de redes en el régimen de muchas fuentes

En el capítulo 5 se estudia la asintótica de muchas fuentes. Este régimen es más adecuado para estudiar un backbone de internet que el e buffer grande. El objetivo de este capítulo se basa en el estudio de los resultados existentes en este régimen para el análisis de performance de un enlace y particularmente para el análisis de una red.

Wischik [44] demostró utilizando grandes desviaciones, un resultado que permite calcular la probabilidad de pérdida en un enlace de la red. Esta expresión ya había sido encontrada por otros autores anteriormente en contextos menos generales [12], [41], [19].

Por otra parte, un resultado reciente [40], para el régimen de muchas fuentes, permite relacionar la distribución del retardo real de los paquetes que arriban a una cola, con la distribución del tamaño de la misma. Este resultado junto al anterior permite calcular la distribución del retardo real en un enlace.

En este régimen una extensión del análisis de un enlace a toda la red fue realizado por Wischik [43]. Wischik demuestra que en el régimen de muchas fuentes la función ancho de banda efectivo (EBW) de un agregado de flujos **independientes** de la salida de un enlace en el límite es igual a la función EBW del agregado de flujos independientes de la entrada al enlace. Esto permite para redes tipo 'in-tree' extender el análisis de un nodo a la red. Sin embargo, no es adecuado para casos de redes más generales y en particular para redes MPLS de un backbone de Internet. Un trabajo reciente de Eun y Shroff [22], demuestran que en el régimen asintótico de muchas fuentes la probabilidad de que la ocupación del buffer sea mayor que 0 tiende a cero casi seguramente. Con este resultado extienden el trabajo de Wischick para un conjunto finito de las infinitas fuentes que utilizan el enlace. Es decir, si se consideran un conjunto finito de fuentes (de las infinitas que alimentan un enlace) la función EBW de ese agregado finito de fuentes a la entrada al enlace y a la salida se mantiene en el límite cuando tiende a infinito la cantidad de fuentes, y la capacidad del enlace y el tamaño del buffer se escalan proporcionalmente. Sin embargo, este resultado

tampoco es adecuado para las necesidades de esta tesis porque para el análisis del siguiente buffer en la red se llega con un número finito de estas fuentes y por tanto no se está en las condiciones del régimen de muchas fuentes en este segundo enlace.

En el capítulo 6 se presenta el marco teórico que a los efectos de esta tesis es el más potente para el análisis de redes MPLS. Este marco se basa en un pre-print reciente de Ozturk, Mazumdar y Likhanov [34]. En este trabajo se considera un régimen algo diferente al de muchas fuentes. En él, las fuentes crecen a infinito con N , al igual que la capacidad, pero el buffer sin embargo cumple que $\lim \frac{B(N)}{N} \rightarrow 0$. En este régimen de buffer pequeño y muchas fuentes, y para topologías muy generales encuentran una relación que permite determinar la función velocidad del gran desvío de la probabilidad de pérdida en un enlace del interior de la red a partir de las funciones velocidad de los diferentes agregados de tráfico en el ingreso a la red. Este resultado permite estimar las pérdidas en cualquier enlace interior de la red. Respecto de la QoS de punta a punta en este trabajo se presentan además dos resultados de interés y que serán parte del marco teórico de este trabajo:

- **La tasa de pérdidas es una métrica cóncava.** Que el logaritmo de la tasa de pérdidas de punta a punta en un camino de la red, es igual (cuando $N \rightarrow \infty$) a menos el ínfimo de las funciones velocidad de la probabilidad de pérdida en cada enlace por los que pasa el camino. Observemos que esto transforma a la tasa de pérdidas en una métrica cóncava, que es una de las cosas que se querían tener para los algoritmos de ingeniería de tráfico.
- **Si se trabaja con la red ficticia no se subestiman las pérdidas** Si se considera una red ficticia donde en cada nodo interior se supone que los agregados de tráfico que arriban a un enlace no fueron afectados por los enlaces anteriores. En esta red ficticia se supone que la estadística de cada agregado no se modifica al pasar por un enlace. Se demuestra en el trabajo de Mazumdar que la región de aceptación de esta

red ficticia está contenida en la región de aceptación de la red real. Es decir que si se acepta un flujo en la red ficticia porque se verifican los requerimientos de QoS (en tasa de pérdidas) en esta red, se puede asegurar que las condiciones de QoS de punta a punta se verifican también en la red real.

Trabajar con la red ficticia mencionada es muy importante. El análisis de la red ficticia simplifica los algoritmos necesarios para evaluar la QoS, que es un aspecto relevante cuando como en esta tesis se pretende trabajar en línea.

Una contribución de esta Tesis y que se desarrolla al final del capítulo 6, es encontrar un conjunto de condiciones que deben cumplirse para que los resultados del análisis de performance en la red ficticia y en la red real coincidan. Esto brinda una condición en la cual se puede analizar la red como si la función EBW del tráfico no se alterara al atravesar un enlace. Estas condiciones permiten evaluar cuando trabajamos con la red ficticia si se comete o no error y la magnitud del mismo.

1.4.3. Estimación de parámetros de QoS

Un problema relevante para poder realizar ingeniería de tráfico en línea es contar con herramientas que permitan medir o estimar los parámetros de QoS sobre una red. En particular será necesario estimar la probabilidad de pérdida o el retardo o aquellos otros parámetros que se consideren importantes en el modelo de trabajo.

Este punto lo abordamos en el capítulo 7. En el contexto de esta tesis, en que se trabaja con grandes desvíos, es necesario estimar el logaritmo de la función generatriz de momentos del tráfico o una función de ésta que es el EBW. Para realizar la estimación de esta función a partir de trazas de tráfico existen dos posibilidades. Utilizar estimadores paramétricos o no paramétricos. En el caso paramétrico se asume algún modelo para el tráfico y los parámetros del modelo se ajustan con la información de las trazas del tráfico. El modelo no paramétrico más usado se basa en estimar el valor esperado de la generatriz de momentos

por un promedio temporal. Este estimador tiene la virtud de ser muy simple, consistente y de uso muy general, pero a partir de él no es fácil calcular un intervalo de confianza para el estimador. En cuanto a los estimadores paramétricos se estudiará un estimador para un modelo markoviano fluido de tráfico. Cuando el proceso de tráfico es modulado por una cadena de Markov en tiempo continuo que decide la tasa de trabajo que genera la fuente, Kesidis et. al. [26] han encontrado una expresión que permite calcular el ancho de banda efectivo, en función del generador infinitesimal de la cadena y de las tasas de generación de tráfico en cada estado. Perera, Simon y Pechiar [35] obtuvieron un estimador del ancho de banda efectivo de la fórmula de Kesidis y calcularon su intervalo de confianza. El cálculo de este estimador para una traza de tráfico real requiere la determinación de los estados de la cadena de Markov y los umbrales que permiten decidir si un punto de la traza pertenece a uno u otro estado de la cadena de Markov.

En el capítulo 7 realizamos una propuesta para ajustar dicho estimador al modelo markoviano 'óptimo' utilizando un test de markovianidad y un algoritmo de decisión.

El otro punto importante del capítulo de estimación es el siguiente. Una vez que se tiene un estimador para el ancho de banda efectivo de la traza de tráfico, surge la pregunta de si será cierto que usando este estimador para calcular las pérdidas en lugar de su valor teórico del ancho de banda efectivo ¿se obtiene un estimador consistente para las pérdidas? ¿en qué condiciones? y si es valido un Teorema Central del Límite para el ancho de banda efectivo ¿será válido un TCL para la probabilidad de pérdida? Estas preguntas fueron abordadas en un trabajo que realizamos junto con otros integrantes del grupo de investigación ARTES (Análisis de Redes Tráfico y Estadística de Servicios)[2]. Los principales resultados de este trabajo y algunas ampliaciones del mismo son descritos en el capítulo mencionado.

1.5. Conclusiones

La tesis aborda tres grandes temas :

- Ingeniería de tráfico y QoS en redes MPLS
- Aplicaciones de Grandes Desvíos al análisis de redes.
- Estimación de tráfico y parámetros de QoS

En este capítulo se ha resumido cómo se enfocaron estos temas y la estrecha relación entre ellos y con los objetivos de la tesis. Además se han reseñado aquellos aspectos donde entendemos se han realizado contribuciones en el desarrollo de esta tesis.

Queremos por último señalar que se ha desarrollado software para estimar y calcular numéricamente varios de algoritmos utilizados en esta tesis. Estos algoritmos fueron implementados en una aplicación JAVA que se reseña en un apéndice.

Capítulo 2

Calidad de Servicio en redes IP

2.1. Introducción

En los últimos años ante el alto grado de difusión de las redes IP ha ido cobrando relevancia la vieja aspiración de contar con una red (ahora IP) donde converjan todos los servicios de telecomunicaciones. Uno de los motores principales de esta 'convergencia', es la necesidad de los distintos operadores de internet de amortizar las inversiones realizadas en infraestructura. Internet fue originalmente pensada como una red académica y en ese sentido el aspecto económico de la red no comenzó a plantearse hasta que se transformó por la vía de los hechos en una red comercial. El paradigma de 'Internet es gratis' ha sido una de las razones que promovieron su expansión, pero cuando la red dejó de estar subsidiada los números de los operadores dejaron de cerrar. Hoy se encuentran con una inversión importante realizada en infraestructura, con una concepción de Internet no adecuada para el cobro de los servicios de transporte de información y con una tasa de crecimiento que si bien existe, se ha ido enlenteciendo.

Un problema central de Internet, es que es muy difícil cobrar por un servicio si no es posible asegurar la calidad adecuada. La política 'best effort' que fue uno de los pilares del crecimiento y la escalabilidad de la red, parte de la premisa de que los paquetes son

entregados a la red y esta tratará de hacerlos llegar al destinatario lo antes posible. Imaginemos lo que sucedería en la red telefónica si el operador le dijera 'Ud. hable que haremos lo posible para que su interlocutor escuche, pero no le aseguramos nada', sin duda pocos pagarían por un servicio así. Los aspectos relacionados con el cobro y la calidad de servicio no adquirieron relevancia real hasta que surgió la necesidad de ofrecer 'nuevos servicios' (telefonía, videoconferencias, etc.) sobre Internet.

Para transferir un archivo o mandar un mail los retardos no son muy relevantes, pero para mantener una conversación telefónica si lo son. Una conversación de buena calidad no es posible con un retardo superior a los 150 ms aproximadamente, y por encima de 300 ms es muy difícil mantener una conversación. Existen también otros servicios que son sensibles a la calidad de servicio por ejemplo las video conferencias, en este caso además del retardo pueden ser relevantes también las pérdidas de paquetes. Como dijimos para los servicios tradicionales el retardo no era un problema y las pérdidas de paquetes se solucionan por retransmisiones implementadas en los protocolos superiores a IP.

En este punto es importante notar que el problema severo de calidad de servicio no se da por el hecho de transmitir voz o video en línea sino por la interactividad del servicio. Es decir, si lo que se pretende es por ejemplo escuchar radio en Internet, se está frente a un servicio de voz en línea. Pero en este caso, el retardo no es relevante porque escuchar la radio con algunos segundos de atraso respecto de la emisión no es demasiado importante. En este caso las pérdidas pueden solucionarse con mecanismos tradicionales y se agrega un buffer en la aplicación destino que pre-memorice algunos segundos de transmisión. Sin embargo, en un servicio interactivo esto no es posible y las retransmisiones no aportan una solución a las pérdidas.

El otro parámetro que aunque es de segundo orden es muy importante en aplicaciones interactivas es el jitter. El jitter es la variación del retardo y tiene importancia porque por ejemplo, la reconstrucción de una señal de voz se debe hacer a la misma cadencia a la que

fue muestreada. Si el retardo de los paquetes es variable esto no es posible. Para solucionar el problema del jitter se puede agregar retardo para absorber esta variabilidad, pero si el servicio es interactivo el margen de maniobra es muy pequeño.

2.2. ¿Donde se generan pérdidas, retardos y jitter en Internet?

Las pérdidas tienen dos causas fundamentales: los errores de transmisión, debido por ejemplo al ruido en el canal de comunicación, y las pérdidas de paquetes en los buffers.

Con la tecnología actual de transmisión, por ejemplo la transmisión óptica, y con algoritmos de recuperación de errores en las capas debajo de IP, las pérdidas debidas a los errores de transmisión son muy poco significativas.

La política 'best effort' implica que si existe más tráfico del que puede ser transportado por un enlace, el sobrante se envía a una cola de donde se irán sacando los paquetes para ser enviados. Si esta cola se llena, los paquetes son descartados.

En cuanto al retardo, el problema es similar. Existen tres fuentes fundamentales de retardo: el retardo de transmisión, el retardo de procesamiento en los enrutadores o switches, y el retardo de las colas de los enlaces. Los dos primeros con la tecnología existente son cada vez menos relevantes. Por ejemplo las transmisiones ópticas llevan a que una transmisión de Uruguay a Europa (asumimos unos 10.000 km) tiene un retardo de transmisión del orden de los 30 ms, es decir 5 0 6 veces menos que lo necesario para mantener una conversación de muy buena calidad. Por otra parte los retardos en enrutadores o switches vienen derivados fundamentalmente del procesamiento de los paquetes. Uno de los procesamientos que genera un retardo es la búsqueda en las tablas de ruteo para decidir el proximo enrutador al que debe enviarse el paquete. Hoy en día estos procesos que antes se hacían por software pueden ser hechos por hardware y las velocidades de procesamiento están llegando al orden del milisegundo por paquete. Por más que el paquete deba atravesar 15 enrutadores esto no le

agregará más de 15 ms en su retardo.

La fuente de retardo que si es muy importante (y además a diferencia de las anteriores es variable de paquete a paquete) es la generada por el tamaño de la cola que encuentra el paquete en cada enlace que atraviesa.

De lo anterior concluimos que el problema fundamental para asegurar calidad de servicio es mantener las colas 'casi vacías'. Ahora bien la pregunta es entonces ¿porqué se llenan las colas? . Las colas se llenan porque la capacidad del enlace es momentáneamente menor que la cantidad de tráfico que pretende usar dicho enlace. Una solución obvia a este problema es asegurar una capacidad tal en todos los enlaces de manera que nunca la velocidad de arribo de paquetes sea mayor que la capacidad del enlace. Pero aquí interviene nuevamente la economía. No es razonable económicamente sobredimensionar toda la red. Pero por otra parte si así lo hiciéramos: ¿por cuanto tiempo estaría sobredimensionada? En estas últimas consideraciones es donde se agregan complicaciones por la naturaleza de IP. Internet es una red fuertemente dinámica en varios aspectos. Por un lado los lugares con alto tráfico cambian. Esto se pudo apreciar por ejemplo con la explosión de los servicios 'peer to peer'. Alguien publica en una máquina en un lugar normalmente con poco tráfico una serie de archivos mp3 o mpeg interesantes y el tráfico hacia esas direcciones aumenta fuertemente. También sucede con eventos especiales, las rutas hacia el servidor que hace hosting del web de FIFA quedaron totalmente congestionadas durante la copa del mundo. Por otra parte, la forma de ruteo IP tradicional (se selecciona el camino con el menor número de saltos) tiende a generar hiperagregación de tráfico en ciertas zonas de la red. Además, ante la caída de un enlace o un cambio en la topología, el camino elegido puede cambiar sustancialmente. Un tercer aspecto es la naturaleza del tráfico. El tráfico es a ráfagas y estas pueden causar congestiones por superposición de las mismas en ciertos enlaces temporalmente.

2.3. ¿Como solucionar los problemas de calidad de servicio y aprovechar los recursos de la red?

Como vimos antes es necesario buscar formas que eviten la congestión en la red. Entendemos por congestión en este contexto la situación en la cual la diferencia entre la tasa de arribo de paquetes y la capacidad del enlace es de tal magnitud o dura un tiempo tal que los requerimientos de calidad de servicio del tráfico (o de una parte de él) no pueden ser satisfechos. La congestión se puede generar porque no se tiene capacidad suficiente para transportar todo el tráfico y satisfacer sus requerimientos de calidad de servicio o porque el tráfico está mal distribuido en la red sobrecargando ciertos enlaces y dejando sub-utilizados otros. Este último punto se resuelve aplicando políticas de Ingeniería de Tráfico. Ingeniería de Tráfico será el tema central del próximo capítulo.

El primer problema que origina la congestión (falta de capacidad) se puede solucionar por dos mecanismos:

- Si la red no tiene capacidad suficiente se debe redimensionar la capacidad de la red. Este problema comúnmente se conoce como planificación de capacidades y no será abordado en esta tesis.
- En muchos casos no se tiene capacidad suficiente para asegurar QoS al total del tráfico. Sin embargo, dentro del tráfico en muchos casos existen diferentes tipos de tráfico con diferentes requerimientos. Si se divide la capacidad de los enlaces separando el tráfico de distintas clases por diferentes 'partes' de la capacidad de cada enlace, se puede lograr cumplir con los requerimientos de QoS de cada clase. Esto se puede lograr a través de la aplicación conjunta de tres mecanismos:
 1. Dividir el volumen total del tráfico en clases con requerimientos diferentes.
 2. Aplicar mecanismos para controlar el volumen de tráfico de cada clase que ingresa la red (con un token bucket por ejemplo)

3. Aplicar políticas de despacho y descarte de paquetes en los enlaces de forma de dividir la capacidad total del enlace en las capacidades necesarias para cumplir los requerimientos de cada clase.

En este último enfoque, se basan modelos de QoS sobre IP como por ejemplo el modelo de Servicios Diferenciados DiffServ. Este será el tema de lo que resta de este capítulo

2.4. Una Arquitectura para ofrecer QoS en IP: DiffServ

La primera arquitectura propuesta para ofrecer QoS en IP fue la arquitectura de Servicios Integrados o IntServ (rfc 1633). Esta arquitectura se basaba en garantizar QoS a través de reservar recursos de punta a punta en la red (de host a host) para cada flujo. Esta arquitectura utiliza el protocolo RSVP (rfc 2205), para efectuar la reserva de recursos y para mantenerla a lo largo de la red. Esta arquitectura si bien garantiza QoS, no es escalable y es impracticable en un backbone del corazón de Internet. Para solucionar el problema de escalabilidad de IntServ en la segunda mitad de la década de los 90 en el IETF comenzó a desarrollarse la arquitectura de servicios diferenciados o DiffServ. Esta arquitectura se basa en dividir el tráfico en clases, controlar la cantidad de tráfico que cada cliente envía a la red de cada clase de tráfico y asegurar requerimientos de QoS utilizando en cada enlace políticas de scheduling y dropping.

En este modelo se establecen acuerdos con el cliente SLA (Service Level Agreements), en el cual entre otras cosas se le garantizan para ciertas clases de tráfico ciertas garantías de QoS siempre que el cliente envíe el tráfico dentro de un cierto perfil (normalmente definido por valores de media, pico y tamaño máximo de burst).

Veremos a continuación brevemente los conceptos básicos de esta arquitectura.

Como mencionamos el tráfico es separado en clases en el ingreso a la red y marcado para registrar la clase a la que pertenece. Esa marca llamada DSCP (Differentiated Service Code Point) usa 6 bits para distinguir una clase de otra. Estos seis bits se registran en

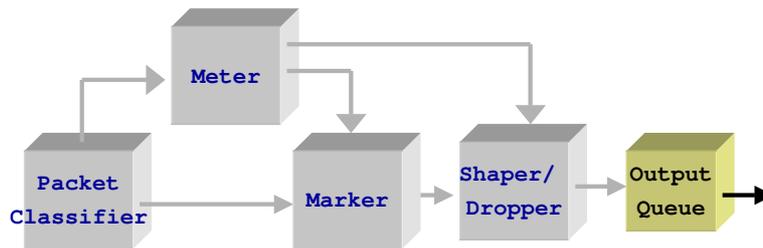


Figura 2.1: Arquitectura de un Nodo exterior en DiffServ

el byte de Type of Service en el cabezal de IPv4 o en el de Traffic Class en el de IPv6. A cada DSCP le corresponderá luego un tratamiento específico en cada nodo de la red. Este tratamiento específico que se le brinda a cada clase de tráfico se llama en DiffServ PHB (Per Hop Behavior). El DSCP es seteado en la frontera de la red y en los routers internos es examinado para asociarlo con el PHB correspondiente. En este sentido la mayor complejidad residirá en los nodos de la frontera, aunque en los nodos interiores habrá que configurar políticas de scheduling y dropping que pueden ser complejas.

En la figura 2.1 vemos la arquitectura de un nodo exterior en Diffserv. Existen dos funciones principales en esta arquitectura:

1. El clasificador, que selecciona paquetes de acuerdo a ciertos criterios y los redirecciona en base a esta selección.
2. El acondicionador de tráfico, que de acuerdo al SLA y en particular al perfil de tráfico acordado, acondiciona el tráfico que ingresa de cada clase.

La clasificación puede ser de dos tipos: MF (MultiField), es decir que analizando diferentes campos del paquete se define la clase a la que pertenece el paquete o simplemente basado en el campo DSCP si el paquete ya venía marcado. El paquete en este modelo puede venir marcado desde el cliente (sea este un usuario final u otro ISP).

La función de acondicionamiento del tráfico clasifica los paquetes en In-profile ó out-of-profile. In-profile puede ser mandado sin ningún otro procesamiento. Los Out-of-profile

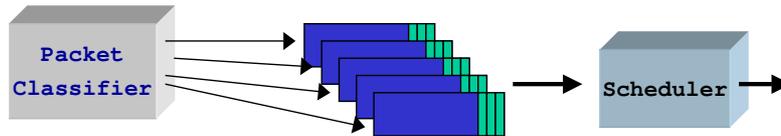


Figura 2.2: Arquitectura de un Nodo interior en DiffServ

podrán ser Re-acondicionados, Re-marcados(en alguna clase más baja por ejemplo) o descartados. Esto dependerá del acuerdo establecido con el cliente.

Los componentes básicos del acondicionador son:

- Meter: realiza mediciones temporales del conjunto de paquetes seleccionados por el clasificador contra el TCA (Traffic conditioning agreement).
- Marker: Setea el campo DS con un código particular y lo asocia así a una clase particular
- Shaper: retarda algunos o todos los paquetes para que cumplan con el traffic profile.
- Dropper: descarta algunos o todos los paquetes para que cumplan con el traffic profile

La arquitectura de un nodo interior de la red se muestra en la figura 2.2.

En esta figura se aprecia que la arquitectura de un nodo interior es algo más simple. En un nodo interior se examina el DSCP y se define el PHB que debe darse al paquete. El PHB está definido como una descripción del comportamiento de reenvío observado exteriormente. Esto quiere decir que en un PHB se especifica como debe observarse como caja negra el tratamiento que reciben los paquetes de esa clase. La implementación de un PHB puede ser hecha por diferentes mecanismos. En general los mecanismos usados actualmente para implementar un PHB son mediante políticas de scheduling para reservar ancho de banda y dropping como RED (Random early detection) o RIO (Red In-profile out-profile).

Los PHBs pueden ser definidos individualmente o como grupo. Un grupo PHB contendrá en general una restricción común como por ejemplo un algoritmo de scheduling

común. Un nodo DiffServ puede soportar múltiples y simultáneos grupos de PHBs. Los recursos serán compartidos entre los grupos de acuerdo a la política de servicios ofrecidos.

Se han definido varios PHBs entre ellos:

- Best Effort. Con un tratamiento similar al de internet actualmente.
- Expedited Forwarding(EF, rfc 2598).El rate mínimo de salida asegurado en todo router al agregado de paquetes EF ,debería ser mayor que el rate máximo de entrada. Para su implementación: se requieren colas con prioridades o WFQ (Weight Fair Queueing), etc. El objetivo es que el flujo agregado vea siempre (o casi) la cola vacía.
- 'Assured Forwarding PHB group' (AF, rfc 2597). Se definen N clases independientes con M niveles de descarte dentro de cada clase. Se utilizan N=4, M=3. A cada clase se le debe asignar una cantidad mínima de recursos y puede obtener más si hay exceso. Dentro de una clase: La probabilidad de un paquete de ser enviado no puede menor si tiene un nivel de descarte menor. Debe responder a condiciones de congestión a largo plazo. Los mecanismos comúnmente usados para su implementación son un Scheduler para reservar recursos y mecanismos de gestión de buffer para manejar niveles de precedencia de descarte.

El PHB especifica cual será el comportamiento que recibirá una clase al atravesar un nodo de la red, pero no dice nada sobre cual será el comportamiento observado de punta a punta de la red. Para esto se define el concepto de PDB (Per Domain Behavior). El concepto de PDB se define en el rfc 3086. La idea es usar PHBs, clasificadores y acondicionadores para componer agregados de tráfico que experimenten un tratamiento especificado cuando transiten por un dominio DiffServ. Especifica métricas para cuantificar el tratamiento que un agregado con un DSCP recibirá al atravesar el dominio.

2.5. Conclusiones

DiffServ brinda una solución a los problemas de escalabilidad que presentaba IntServ. Esto lo logra a través de dar garantías a agregados de flujo y no a flujos individuales. Sin embargo presenta algunos inconvenientes. Si en virtud del ruteo de paquetes se produce hiperagregación de tráfico en algunas zonas de la red, puede degradarse el servicio incluso en las clases más altas y aún cuando en la red existan zonas poco cargadas. Por otra parte si bien la arquitectura de los nodos interiores es más simple que la de los nodos de la frontera, en los nodos interiores se requiere implementar políticas de scheduling y dropping relativamente complejas y por tanto la configuración de ellos no será tan simple como en una red IP tradicional. Esto trae problemas de operación y gestión de estas redes. También trae problemas de escalabilidad ya que el agregado de un nodo interior o el cambio de la topología del interior de la red puede requerir la reconfiguración de schedulers y droppers en diversos nodos de la red. Basados en estas debilidades se buscarán mecanismos que complementen o simplifiquen algunos aspectos de esta arquitectura.

Capítulo 3

Ingeniería de Tráfico en MPLS

3.1. Introducción: Conceptos básicos de la arquitectura MPLS

Multi Protocol Label Switching (MPLS) surgió en los últimos años de la década de los 90 como una arquitectura que debiera permitir mejorar la performance de las redes IP. Sin embargo, actualmente su interés radica en sus aplicaciones a redes privadas virtuales, a Ingeniería de Tráfico y a QoS sobre IP.

La arquitectura de una red MPLS está definida en el rfc 3031. En MPLS los enrutadores se llaman LSR (label switched router) y los que se encuentran en la frontera de la red se llaman LER (Label Edge router).

En una red MPLS se hace una partición del conjunto de todos los posibles paquetes que ingresan a la red en clases de equivalencia denominadas 'Forwarding Equivalence Classes (FECs)'. Una FEC es un grupo de paquetes que serán re-enviados de la misma manera en el dominio MPLS. En IP tradicional es el 'longest match prefix' quien determina el re-envío, en MPLS podrá ser este el criterio u otro. Una vez asignados a una FEC los paquetes son indistinguibles desde el punto de vista de su reenvío. En MPLS la FEC se determina en la entrada y no en cada router. Paquetes con distinto destino pueden agruparse en la misma FEC. Posibles criterios de FEC son: IP Destino, IP Origen – IP Destino, IP Origen - IP Destino - puerto origen – puerto destino, Etc.

A cada FEC se le asocia una etiqueta en el nodo de ingreso y se envía el paquete al nodo siguiente con la etiqueta correspondiente. La etiqueta es local, es decir que es un valor arbitrario acordado entre dos vecinos en el camino de un paquete. Cada nodo interior leerá la etiqueta cuando ingresa el paquete y en función de ella (o de ella y la interfaz de entrada) definirá la interfaz de salida y la etiqueta con que se enviará el paquete al próximo nodo. El último nodo de la red para un paquete, eliminará la etiqueta y enviará el paquete hacia fuera de la red MPLS tal como arribó al dominio. Puede acordarse también entre el último y el penúltimo nodo que sea el penúltimo quien retire la etiqueta, con lo cual puede evitarse en el último nodo dos búsquedas en las tablas de envío (primero en la tabla de MPLS y luego en IP).

Una vez que entre todos los nodos por los que circulará una cierta FEC se estableció un acuerdo de cual será la etiqueta de envío a cada nodo, se establece un camino virtual denominado LSP en MPLS (Label switched path).

La arquitectura MPLS permite manejar más de un nivel de etiquetas. Es decir que se podrá asociar a un paquete un stack de etiquetas. Esto tiene importancia por diferentes razones, en particular a los efectos de poder establecer una red jerárquica y para realizar agregación de LSPs en determinadas zonas de la red.

La forma de encapsular la etiqueta tiene dos modalidades básicas. A través de un encapsulado genérico definido en la rfc 3032 o a través de un cabezal de capa 2 por ejemplo encapsulando la etiqueta en ATM a través del VCI/VPI o en Frame Relay a través del DLCI.

Para establecer las etiquetas para un LSP, es necesario un protocolo de distribución de etiquetas. Existen diversas propuestas de protocolos para distribución de etiquetas como por ejemplo: LDP (RFC 3036), Carrying Label Information in BGP-4 (RFC 3107), RSVP-TE: Extensions to RSVP for LSP Tunnels (RFC 3209), Constraint-Based LSP Setup using LDP (RFC 3212). No se analizará en detalle estos protocolos ya que no son relevantes

para entender la tesis. En la sección referida a ingeniería de tráfico se realizarán algunos comentarios sobre la utilización de estos protocolos para realizar ingeniería de tráfico.

3.2. Ingeniería de tráfico en MPLS

Ingeniería de tráfico es el proceso de mapear la demanda de tráfico sobre la topología de la red. Es la habilidad de controlar el flujo de tráfico sobre la red.

El RFC 2702, 'MPLS Traffic Engineering (TE)' establece que la ingeniería de tráfico concierne a la optimización de la performance de una red e involucra diversas áreas: Mediciones de tráfico, Modelado de tráfico y redes, Control del tráfico en Internet, Evaluación de performance.

Se establece que los principales Objetivos de TE son:

- Mover el tráfico del camino establecido por el IGP (Interior Gateway Protocol) a un camino menos congestionado
- Utilizar el exceso de ancho de banda sobre los enlaces sub-utilizados
- Maximizar la utilización de los enlaces y nodos de la red.
- Aumentar la confiabilidad del servicio
- Alcanzar requerimientos impuestos

Los requerimientos pueden ser: Orientados al tráfico: pérdidas de paquetes, retardos, etc. u orientados a los recursos: fundamentalmente utilización de la capacidad de la red.

Las acciones de control tomadas al realizar TE pueden involucrar: a. Modificación de los parámetros de Gestión de Tráfico

b. Modificación de los parámetros asociados al ruteo

c. Modificación de los parámetros y atributos asociados con los recursos.

En general se busca también minimizar la intervención manual para tomar acciones de control.

Se utilizará en diversos puntos de la tesis el concepto de troncal de tráfico. Una troncal de tráfico en este contexto es un agregado de flujos pertenecientes a la misma clase. En modelos con una sola clase, puede encapsular todo el tráfico entre dos LERs. Las troncales son objetos enrutables y deben diferenciarse del LSP que utiliza la troncal en un momento dado. Esta distinción es importante porque el LSP puede cambiar pero la troncal sigue siendo la misma. Al igual que el LSP la troncal es unidireccional.

La ingeniería de tráfico debe resolver tres problemas básicos:

- Cómo mapear paquetes en FECs
- Cómo mapear FEC en troncales tráfico.
- Cómo mapear troncales en la red física

3.2.1. Ruteo Explícito

La principal característica de MPLS que permite realizar TE es el ruteo explícito. Una ruta explícita es una secuencia de nodos lógicos entre un nodo de ingreso y uno de egreso que se definen y establecen desde un nodo de la frontera. Una ruta explícita puede ser una lista de direcciones IP. También pueden especificarse los primeros N saltos solamente y luego la ruta definida por el protocolo de ruteo IP. Puede usarse también en una ruta explícita el concepto de Nodo Abstracto: Colección de nodos presentados como un solo paso en una ruta explícita. Un ejemplo de nodo abstracto puede ser un Sistema Autónomo.

Si el nodo ingreso quiere establecer una ruta que no sigue el camino que sigue por defecto el protocolo de ruteo IP, debe utilizar un protocolo de distribución de etiquetas que soporte la definición de rutas explícitas. Existen dos definidos por el IETF: CR-LDP y RSVP.

La ruta LSP puede ser restringida por la capacidad de recursos y la capacidad de los nodos de cumplir con los requerimientos de QoS. Esto lleva al concepto de “constrained

route” (CR) o ruta con restricciones. Una ruta con restricciones es una ruta que se obtiene imponiendo un conjunto de restricciones que se deben cumplir. Por ejemplo: información de QoS del enlace (ancho de banda disponible, retardo, etc.), clases, prioridades, etc.

EL LSR de ingreso calcula una ruta que satisfaga un conjunto de restricciones en el estado actual de la red.

Para encontrar una CR se debe correr un algoritmo de ruteo basado en restricciones (Constrained base routing).

En el rfc 2702 se establece que para realizar TE una red MPLS y particularmente CBR debe ser posible definir:

- Atributos asociados a las troncales de tráfico que en conjunto especifican su comportamiento. Dentro de estos atributos se encuentran:
 - Parámetros del tráfico de la troncal. Caracterización del tráfico que utilizará esa troncal.
 - Atributos para el establecimiento y mantenimiento de caminos establecidos administrativamente
 - Reglas para establecer preferencias de ciertos caminos que pueden ser mandatorios o no. Se considera adecuado tener atributos que establezcan una jerarquía o preferencia para mapear una troncal dentro de un conjunto de posibles caminos.
 - Clase de afinidad con recursos. Se recomienda contar con atributos que permitan establecer clases de afinidad entre los recursos y las troncales de forma de establecer caminos para las troncales usando aquellos recursos que le son afines.
 - Adaptabilidad a cambios. Debe poder especificarse si ante cambios en el estado de la red se re-calculan o no los caminos establecidos para la troncal.
 - Prioridad y Preemptividad de las diferentes troncales a la hora de establecer y de mantener un LSP.

- Atributos asociados al re-enrutamiento. Se debe poder decidir ante cambios en la red, si una troncal se re-enrutará solo si hay caminos con recursos suficientes o si se re-enrutará siempre.
 - Atributos de Policing para definir qué acciones se toman si la troncal no cumple con los parámetros de tráfico que se especificaron.
- Atributos asociados a los recursos. En este punto se establecen básicamente dos atributos
 - Maximun Allocation Multiplier Máximo ancho de banda que se permite reservar para los caminos que atraviesan dicho enlace
 - Clases de recursos. Afinidad que restringe el mapeo de las troncales sobre los recursos.
 - Constrained base routing para realizar el mapeo. Este algoritmo tendrá en cuenta: Atributos asociados con las troncales de tráfico, atributos asociados con los recursos e información del estado de la red.

Propuestas e implementaciones de CBR

Existen diversas propuestas académicas de algoritmos para realizar CBR. Un resumen de algunas de ellas se encuentra en el trabajo [31]. Uno de los principales problemas que enfrenta el ruteo basado en restricciones es que casi cualquier problema de interés es NP-completo. Por ejemplo en el trabajo de Wang et al. [45] se clasifican las métricas de interés para el ruteo basado en restricciones de la siguiente manera:

Sea $d(i, j)$ una métrica del enlace (i, j) . Para todo camino $p = (i, j, k, \dots, l, m)$, decimos que la métrica es

$$\text{aditiva si: } d(p) = d(i, j) + d(j, k) + \dots + d(l, m)$$

$$\text{cóncava si: } d(p) = \min[d(i, j), d(j, k), \dots, d(l, m)]$$

multiplicativa si $d(p) = d(i, j)d(j, k)...d(l, m)$

Por ejemplo el Retardo, el jitter y el costo son métricas aditivas, el ancho de banda es cóncava y las pérdidas se pueden transformar en una métrica multiplicativa. En dicho trabajo se demuestra que encontrar un camino en una red sujeto a restricciones en dos o más métricas aditivas o multiplicativas y en cualquier combinación de ellas es NP-completo.

A raíz de este problema se han propuesto numerosos algoritmos heurísticos, varios de ellos detallados en [31], para encontrar caminos sujetos a un conjunto de restricciones. Algunos de estos algoritmos son aptos solo para trabajar fuera de línea ya que por su complejidad se dificulta su implementación en línea. Otros que se proponen para aplicaciones en línea como las que son de interés en la tesis, tienen la dificultad que se basan en el conocimiento de determinados parámetros de la red que no son fáciles de implementar. Por ejemplo algunos se basan en el conocimiento de las pérdidas de punta a punta en una red real. Esto es muy difícil de medir pérdidas en una red ya que los valores son tan bajos en un backbone real que tener una estadística suficiente no es un problema que se pueda resolver para aplicar en un algoritmo en línea. Otras requieren mediciones del retardo 'one-way' en los diferentes caminos. Medir el retardo 'one-way' en una red asíncrona no es posible. Es posible estimarlo a partir del retardo de ida y vuelta, pero si se puede medir el retardo ida y vuelta, ¿cuánto es el retardo 'one-way'? obviamente la respuesta dependerá de lo cargado o no que se encuentre el camino que queremos medir y el de vuelta. Además medir el retardo de ida y vuelta también presenta problemas no obvios de resolver por ejemplo: ¿cuánto es la cantidad mínima de paquetes de prueba necesarios para tener una estadística suficiente?. Varios de estos problemas están aún abiertos y en el IETF existe un grupo (IPPM) trabajando sobre ciertas normalizaciones en este punto.

La realidad muestra que las implementaciones de CBR en línea existentes en la práctica, se basan en el algoritmo CSPF (Constrained Shortest Path First) [21] [27]. CSPF es un algoritmo basado en el conocido algoritmo SPF. CSPF utiliza en general como restricción

el ancho de banda requerido del LSP a establecer y lo compara con el ancho de banda disponible en los enlaces (ancho de banda reservable del enlace menos el ancho de banda ya reservado). Con este criterio primero se podan todos aquellos enlaces que no satisfagan la restricción de ancho de banda. Luego ,si existen otros criterios administrativos, como asignaciones de afinidad entre enlaces y el LSP que se desea establecer, se podan aquellos enlaces que no satisfagan estas restricciones administrativas. Sobre este 'árbol podado' se aplica el algoritmo SPF para encontrar el camino más corto (teniendo en cuenta los pesos de los enlaces) que conducen del LER de ingreso al LER de egreso. La utilización del ancho de banda como restricción no es suficiente para tráfico con requerimientos de QoS como voz o video, excepto que se reserven los picos máximos para todos los agregados de tráfico, pero en este caso no se estaría aprovechando las ventajas del multiplexado estadístico. Una de las propuestas de esta tesis es modificar el algoritmo CSPF para satisfacer requerimientos de QoS de diferentes clases de tráfico

Además del algoritmo de CBR que defina el camino, una propuesta de CBR requiere dos componentes más. Un protocolo que permita establecer los caminos una vez definidos y un protocolo que difunda información del estado de la red. Para el primer punto hay propuestos dos protocolos y ambos tienen implementaciones en diferentes enrutadores comerciales: RSVP-TE y CR-LDP. Para el segundo problema el protocolo más difundido que resuelve estos aspectos son versiones extendidas de los protocolos OSPF o IS-IS. Estas extensiones aún son propietarias de algunos fabricantes o se encuentran en estado de draft en el IETF, un ejemplo se puede ver en [42]. En la propuesta de ingeniería de tráfico se utilizarán estos mismos protocolos pero se difundirá como información de estado, otra información que no es la del algoritmo estándar CSPF.

3.2.2. Reparto de carga

El RFC 3031 ([39]), habilita a realizar balanceo o reparto de carga entre diferentes LSPs. Balancear carga es una potente herramienta de ingeniería de tráfico en MPLS. Esta herramienta brinda la posibilidad de enrutar troncales cuyo tráfico es superior a las posibilidades de un único camino en la red, y permite también mejorar el uso de recursos de la red.

Para realizar reparto de carga se deben tener en cuenta dos aspectos. El primero es el algoritmo con el cual se decide los coeficientes de reparto de carga entre los LSPs. El segundo es el mecanismo (una vez fijados los coeficientes) que se utiliza para asignar los paquetes a uno u otro LSP.

Respecto de los algoritmos para balancear carga en una red MPLS existen diferentes propuestas y que resumiremos en la sección siguiente.

Sobre los mecanismos para realizar el reparto de carga, esto no ha sido aún estandarizado por el IETF. A la fecha en que se escribe esta tesis, existe en discusión dentro del IETF un draft ([1]) que aborda parte de estos problemas.

Propuestas e implementaciones de reparto de carga en MPLS

Varios autores han abordado el problema del reparto de carga entre LSPs, que es una de las herramientas que habilita usar la arquitectura MPLS. En estos casos se supone que ya se encuentran configurados un conjunto de LSPs y el problema que se plantea es para cada agregado de flujo en cada LER cómo repartir su carga entre los diferentes LSPs asociados a dicho agregado. Algunos autores (por ej. [20]) proponen algoritmos en línea para repartir carga (MATE) y otros (por ej. [37]) han propuesto mecanismos fuera de línea para esto. En ambos casos se busca optimizar alguna medida de performance como por ejemplo las pérdidas promedio o el retardo medio en la red. MATE es un algoritmo potente pero asume que el tráfico es constante en media y además es muy sensible a las variaciones del parámetro

que se debe medir en línea para ajustar el reparto. Por ejemplo en la propuesta realizada en el trabajo [20] se debe medir la derivada del retardo con respecto a la capacidad de cada enlace y para tal fin se asumen modelos de los enlaces simples (M/M/1). MATE no garantiza requerimientos de QoS al tráfico, simplemente optimiza un costo global de la red, como el retardo promedio, pero nada garantiza que para cada agregado se puedan garantizar sus restricciones de QoS. Se puede tener un retardo promedio en la red mínimo y para algunos agregados no ser posible cumplir con sus requerimientos de QoS.

Los algoritmos fuera de línea obtienen formas de cálculo que son adecuadas para una optimización de largo plazo de la red pero que son de difícil aplicación en escalas más cortas. Por otra parte en su aplicación real cabe preguntarse que quiere decir "óptimos" en el largo plazo, cuando la red sufre variaciones de diferente índole y estos algoritmos se basan en estimaciones estadísticas de diversos parámetros. Este tipo de algoritmos además tratan de optimizar algún parámetro de la red, por ejemplo que las pérdidas máximas en la red sean mínimas, pero como dijimos, esto puede no lograr una configuración donde cada agregado de flujo tenga las pérdidas que se requieren para cumplir sus requerimientos de QoS. Es decir, puede convenir que las pérdidas en algunos agregados sean mayores, pero gracias a eso disminuir las pérdidas para el tráfico más restrictivo en cuanto a la QoS.

Una vez definidos los coeficientes de reparto de carga, existen dos formas de repartir carga entre LSPs:

- por paquete
- por flujo

El primer modo es más simple de implementar y más preciso. El segundo es más complejo de implementar y si no se tienen muchos flujos o estos son muy diferentes, es poco preciso. Sin embargo, este último método presenta una fuerte ventaja al mantener el ordenamiento de los paquetes de un flujo.

Veremos ahora algunos comentarios sobre las implementaciones de reparto de carga en enrutadores comerciales [21], [27]. Actualmente en los enrutadores comerciales se puede realizar reparto de carga pero no se puede ajustar en línea los coeficientes de reparto. Es decir, se configuran los coeficientes de reparto y luego estos quedan fijos no existiendo un mecanismo de ajuste si hay variaciones de tráfico. Lo que sucede por defecto, es que si se establece más de un LSP para una troncal, la carga se reparte por partes iguales entre los LSPs establecidos. En algunos casos se implementa una variante algo más sofisticada que es repartir carga inversamente proporcional al ancho de banda reservado por cada LSP. En otros enrutadores se permite configurar coeficientes de reparto de carga y repartir de acuerdo a estos coeficientes. En general también se permite configurar reparto de carga por paquete o por flujo.

3.3. Conclusiones

En este capítulo hemos resumido las características generales de MPLS. También se analizó MPLS como arquitectura para realizar ingeniería de tráfico. Se analizaron en particular dos herramientas de TE, Ruteo basado en restricciones (CBR) y Reparto de carga. Se analizaron las características de estas herramientas, algunas propuestas existentes y se vieron las características generales de lo que se puede encontrar hoy en enrutadores comerciales.

Como hemos mencionado antes, el objetivo de la tesis es hacer ingeniería de tráfico en línea y en particular buscar algoritmos que permitan asegurar requerimientos de QoS de punta a punta en una red. Por este motivo en los próximos cuatro capítulos se estudiarán los mecanismos existentes para evaluar performance de punta a punta en una red y las herramientas que permitan realizar estimaciones de los parámetros de QoS en una red. En el último capítulo veremos como adaptar algunas de las herramientas de TE vistas en este capítulo para cumplir con los objetivos de la tesis.

Capítulo 4

Grandes desviaciones aplicadas al análisis de performance en la asintótica de "buffer grande"

4.1. Introducción

En la aplicación de la teoría de grandes desviaciones al análisis de redes de Telecomunicaciones se ha trabajado sobre dos asintóticas: la asintótica de 'buffer grande' y la asintótica de 'muchas fuentes'. En el primer caso se estudia el comportamiento 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 como se muestra en la figura 4.1.

En el segundo se estudia el comportamiento del 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 4.2).

El objetivo de la tesis es analizar el comportamiento de una red que opera en el corazón de Internet, un backbone nacional o regional con mucho tráfico. En estas redes, es donde más se justifica el uso de herramientas de análisis de performance y de ingeniería de tráfico para asegurar calidad de servicio (retardo, pérdidas, jitter) y hacer el mejor uso posible de sus recursos (capacidad, buffer, etc.). En este tipo de redes es mucho más razonable la hipótesis de 'infinitos flujos' que la de 'buffer grande'. Esto se debe a que en cada enlace confluyen

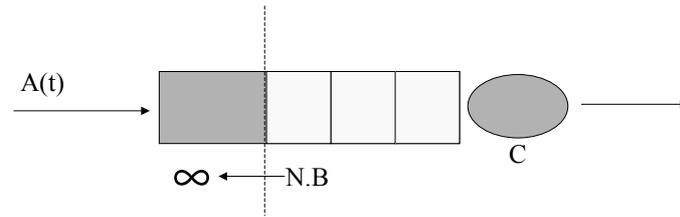


Figura 4.1: Régimen asintótico de buffer grande

una gran cantidad de agregados de flujos y el buffer por fuente en general es pequeño. El buffer en dichas redes se dimensiona para atender la simultaneidad en la llegada de paquetes, ya que no es en general razonable colocar un buffer lo suficientemente grande como para atender simultaneidad de ráfagas de una cantidad de flujos muy grandes. Por esta razón el foco principal estará en la asintótica de muchas fuentes, pero antes se revisarán algunos resultados del régimen de buffer grande que ayudarán a entender algunos problemas importantes

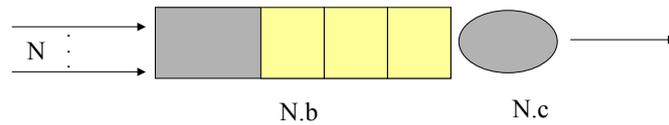


Figura 4.2: Régimen asintótico de muchas fuentes

4.2. 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 $C(n), n \in \mathbb{Z}$. Llamaremos $Q(n), n \in \mathbb{Z}$ al tamaño de la cola. Definiremos $X(n) = A(n) - C(n)$. 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:

$$Q(n+1) = (Q(n) + X(n))^+ \quad (4.2.1)$$

donde Z^+ es el máximo entre Z y 0 .

A lo largo del trabajo se analizará fundamentalmente el caso de tiempo discreto, y se generalizará cuando presente interés particular a tiempo continuo, aunque la mayoría de los resultados que exponemos son válidos en ambos contextos. De la ecuación de Lindley podemos escribir

$$Q(0) = \text{máx}(X(-1) + Q(-1), 0) \quad (4.2.2)$$

$$= \text{máx}(X(-1) + \text{máx}(X(-2) + Q(-2), 0), 0) \quad (4.2.3)$$

$$= \text{máx}(X(-1), X(-1) + X(-2) + Q(-2), 0) \quad (4.2.4)$$

$$= \text{máx}(X(-1), X(-1) + X(-2), \dots, \sum_{i=-n}^{i=-1} X(i) + Q(-n), 0) \quad (4.2.5)$$

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

$$Q(0) = \text{máx}(S_0, S_1, \dots, S_n + Q(-n)) \quad (4.2.6)$$

La pregunta que surge es si cuándo $n \rightarrow \infty$ este proceso tendrá límite. Es decir, en que casos el trabajo en la cola 'en régimen' tendrá una distribución estacionaria y cuándo esta 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 4.1. *Teorema de Loynes (ver su demostración por ejemplo en [4], pag. 74). Si se cumple que*

- *El proceso X es estacionario*
- *El proceso X es ergódico*
- $\mathbf{E}(X) < 0$

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

$$\lim_{n \rightarrow \infty} \mathbb{P}(Q(n) \leq x) = \mathbb{P}(\sup_{n > 0} S_n \leq x) \quad (4.2.7)$$

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

Este resultado será extensamente usado en las próximas secciones.

4.3. Definiciones básicas de la Teoría de Grandes Desviaciones y Teorema de Crámer

La teoría de grandes desviaciones se centra en el estudio asintótico del comportamiento de eventos raros". Por ejemplo si en un enlace de una red de datos, el tamaño del buffer tiende a infinito, la probabilidad de pérdida tiende a 0, la teoría de grandes desviaciones dirá con qué velocidad tiende a 0. En esta sección se brindarán las principales definiciones a usar en el resto de este trabajo. A lo largo de las restantes secciones, en la medida que sean necesarios para las aplicaciones a Redes de Telecomunicaciones, se enunciarán los

teoremas fundamentales de la Teoría de Grandes Desviaciones y se brindarán las principales ideas y consideraciones de la demostración para que el lector pueda seguir el desarrollo. La demostración detallada de estos teoremas se puede encontrar en el libro de Dembo y Zeitouni [18], ó en las referencias que se citarán.

Sea μ_n una familia de medidas de probabilidad sobre un espacio (χ, B)

Una **función de velocidad** (rate function, *R.F.*) es una función semicontinua por debajo (lower semicontinuos, *L.S.*) $I : \chi \rightarrow [0, \infty)$ (*L.S.* son las funciones cuyos conjuntos de nivel son cerrados).

Sea $\Psi_\alpha(I) \triangleq \{x : I(x) \leq \alpha\}$ el conjunto de nivel α de I . Una función I es una **buena función de velocidad** (good rate function, *G.R.F.*) si todos los conjuntos de nivel de I son conjuntos compactos de χ .

Llamaremos dominio efectivo de I $D_I^o \triangleq \{x : I(x) < \infty\}$

Definición 4.1. μ_n *satisface el principio de las grandes desviaciones(LDP) con R.F., I si* $\forall \Gamma \in B$

$$-\inf_{x \in \Gamma^o} I(x) \leq \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(\Gamma) \leq \limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(\Gamma) \leq -\inf_{x \in \bar{\Gamma}} I(x) \quad (4.3.1)$$

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

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

Definición 4.2. μ_n *satisface el principio de las grandes desviaciones débil con R.F., I, si la cota superior de 4.3.1 se verifica para todo compacto y la cota inferior se verifica para todo abierto.*

Un ejemplo de medida que cumple un LDP débil pero no un LDP es el caso que μ_n sea la medida degenerada en n .

Definición 4.3. *Una familia de medidas μ_n en \mathcal{X} es de decaimiento exponencial (exponentially tight) si para todo $\alpha < \infty$ existe un compacto K_α tal que*

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(K_\alpha^c) \leq -\alpha \quad (4.3.2)$$

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

Definición 4.4. Sea X_1, \dots, X_n, \dots una sucesión de variables aleatorias i.i.d. en el espacio \mathbb{R}^d y sea μ su ley de probabilidad.

Llamamos función logarítmica generadora de momentos de μ a la función

$$\Lambda(\lambda) \triangleq \log(M(\lambda)) \triangleq \log(E(\exp^{\langle \lambda, X_1 \rangle})) \quad (4.3.3)$$

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

Definición 4.5. Llamaremos transformada de Fenchel-Legendre de $\Lambda(\lambda)$ a la función

$$\Lambda^*(x) \triangleq \sup_{\lambda \in \mathbb{R}^d} \{\langle \lambda, x \rangle - \Lambda(\lambda)\} \quad (4.3.4)$$

4.3.1. Teorema de Cramér

Teorema 4.2. Teorema de Cramér. Sea \overline{S}_n con ley μ_n definida de la siguiente manera

$$\overline{S}_n = \frac{1}{n} \sum_{i=1}^n X_i \quad (4.3.5)$$

entonces μ_n satisface un LDP débil con función de velocidad convexa $\Lambda^*(x)$. Si además $0 \in D_\Lambda^o$, entonces μ_n satisface un LDP con 'buena función de velocidad' $\Lambda^*(x)$.

La prueba en \mathbb{R}^d realizada por Dembo y Zeitouni en capítulo 2 de [18] (que es la que se esbozará a continuación), exige que la función logarítmica generadora de momentos $\Lambda(\lambda)$ tenga dominio efectivo D_Λ^o en todo \mathbb{R}^d . Sin embargo a partir de la extensión de este teorema a espacios topológicos más generales en el capítulo 6 de dicho libro se prueba que solo es necesaria la condición de que $0 \in D_\Lambda^o$. La idea de la demostración desarrollada por Dembo y Zeitouni para \mathbb{R}^d se centra en los siguientes puntos:

1. Cota superior. En este caso la demostración se basa en aplicar la desigualdad de Chebycheff para bolas abiertas en \mathbb{R}^d . Se puede probar que establecer la cota superior de un LDP, es equivalente a probar que se verifica para cada $\delta > 0$ y todo conjunto cerrado F que:

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(F) \leq \delta - \inf_{x \in F} I^\delta(x) \quad (4.3.6)$$

siendo $I^\delta(x) \triangleq \min(I(x) - \delta, \frac{1}{\delta})$.

Para probar la cota superior se prueba primero para compactos. Sea un conjunto compacto $\Gamma \subset \mathbb{R}^d$. Dado δ , para cada $q \in \Gamma$ se elige $\lambda_q \in \mathbb{R}^d$ para el cual $\langle \lambda_q, q \rangle + \Lambda(\lambda_q) \geq I^\delta(q)$ y se elige ρ_q tal que $\rho_q |\lambda_q| \leq \delta$.

Aplicando la desigualdad de Chebycheff se puede ver que:

$$\mu_n(B(q, \rho)) = \mathbf{E}(\mathbf{1}_{\overline{S_n} \in B(q, \rho)}) \leq \quad (4.3.7)$$

$$\mathbf{E}[\exp(n\langle \lambda, \overline{S_n} \rangle)] \exp(- \inf_{x \in B(q, \rho)} \{n\langle \lambda, x \rangle\}) \quad (4.3.8)$$

luego,

$$\frac{1}{n} \log \mu_n(B(q, \rho)) \leq - \inf_{x \in B(q, \rho)} \{n\langle \lambda, x \rangle\} + \Lambda(\lambda) \quad (4.3.9)$$

Por lo tanto de 4.3.9 se puede ver que

$$\frac{1}{n} \log \mu_n(B(q, \rho)) \leq \delta - \langle \lambda_q, q \rangle + \Lambda(\lambda_q) \quad (4.3.10)$$

lo que prueba la cota superior para bolas centradas en cualquier punto $q \in \Gamma \subset \mathbb{R}^d$:

$$\frac{1}{n} \log \mu_n(B(q, \rho)) \leq \delta - I^\delta(q) \quad (4.3.11)$$

Posteriormente como todo compacto en \mathbb{R}^d puede ser cubierto con una colección numerable de abiertos, se prueba la cota superior para compactos. Para extenderlo a cerrados, se prueba que la ley μ_n es 'exponentially tight', usando que el dominio efectivo de Λ es todo \mathbb{R} . Como mencionamos, esta última hipótesis no es estrictamente necesaria y solo es necesario que $0 \in D_\Lambda^o$.

2. Cota inferior. En este caso se prueba que para cualquier bola centrada en $y = \nabla(\Lambda(\eta))$ para algún punto $\eta \in \mathbb{R}^d$ se verifica la cota inferior. Para esto define una nueva medida $\frac{d\tilde{\mu}}{d\mu}(z) = \exp(\langle \eta, z \rangle - \Lambda(\eta))$ trasladando la media de la medida original al punto y en cuestión. Luego $\forall \delta$ y para $0 < \varepsilon < \delta$ se realiza la siguiente acotación $\mu_n(B(y, \delta)) \geq \mu_n(B(y, \varepsilon)) \geq \exp(-n\varepsilon|\eta|) \exp(n\Lambda(\eta)) \tilde{\mu}(B(y, \varepsilon))$. Posteriormente aplicando la ley de los grandes números se ve que $\tilde{\mu}_n(B(y, \varepsilon)) \rightarrow 1$ y se verifica la tesis del teorema. En esta parte de la demostración hace uso extensivo del hecho de que el dominio de Λ es todo \mathbb{R} , lo cual como se mencionó no es estrictamente necesario.

Ejemplos y comentarios:

1. La transformada de Fenchel-Legendre puede interpretarse como se muestra en la figura 4.3. La distancia entre la recta λx y la tangente a $\Lambda(\lambda)$ con pendiente x es la transformada $\Lambda^*(x)$.

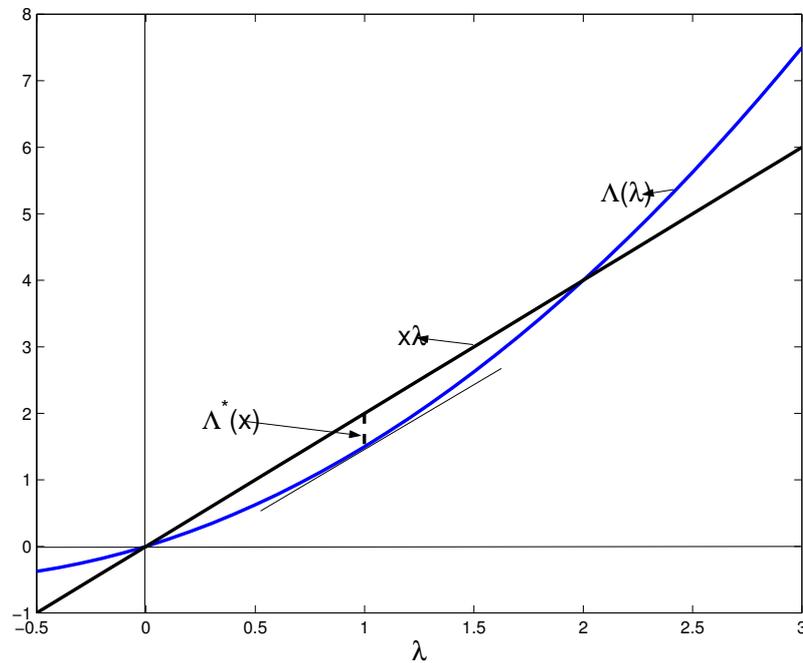


Figura 4.3: Transformada de Fenchel-Legendre

2. Para v.a. acotadas es decir que existe algún K tal que

$$\mathbb{P}(|X_i| > K) = 0 \quad (4.3.12)$$

el dominio efectivo de Λ 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

$$\Lambda(\lambda) = \log\left(\frac{\exp(\lambda) + 1}{2}\right) \quad (4.3.13)$$

y su transformada de Fenchel-Legendre

$$\Lambda^*(x) = \log 2 + x \log(x) + (1-x) \log(1-x) \quad 0 \leq x \leq 1 \quad (4.3.14)$$

$$\Lambda^*(x) = \infty \text{ en otro caso} \quad (4.3.15)$$

3. Variables Gaussianas independientes con media μ y varianza σ^2 . En este caso se obtiene la siguiente FLGM y su transformada de F-L

$$\Lambda(\lambda) = \mu\lambda + \frac{\lambda^2\sigma^2}{2} \quad (4.3.16)$$

$$\Lambda^*(x) = \frac{(x-\mu)^2}{2\sigma^2} \quad (4.3.17)$$

En este caso el dominio efectivo de Λ son todos los reales, al igual que en el caso anterior.

4. Variables exponenciales independientes de parámetro μ . En este caso se obtiene la siguiente FLGM

$$\Lambda(\lambda) = \log\left(\frac{\mu}{\mu-\lambda}\right) \quad \lambda < \mu \quad (4.3.18)$$

$$\Lambda(\lambda) = \infty \quad \lambda \geq \mu \quad (4.3.19)$$

y su transformada de Fenchel-Legendre

$$\Lambda^*(x) = \mu x - 1 - \log(\mu x) \quad x > 0 \quad (4.3.20)$$

$$\Lambda^*(x) = \infty \quad x \leq 0 \quad (4.3.21)$$

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

5. Variables con distribución de Poisson con parámetro μ . En este caso

$$\Lambda(\lambda) = \mu(\exp(\lambda) - 1) \quad (4.3.22)$$

y su transformada de FL es:

$$\Lambda^*(x) = x(\log(\frac{x}{\mu}) - 1) + \mu \text{ si } x > 0 \quad (4.3.23)$$

$$\Lambda^*(x) = \mu \text{ si } x = 0 \quad (4.3.24)$$

$$\Lambda^*(x) = \infty \text{ si } x < 0 \quad (4.3.25)$$

6. Variables independientes con distribución de Cauchy. En este caso $\Lambda(\lambda) = 0$ para $\lambda = 0$ y vale ∞ para todos los demás valores de λ 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 veremos a continuación

4.3.2. Teorema de GARTNER -ELLIS

Sea una sucesión de variables aleatorias Z_n en \mathbb{R}^d con medida μ_n . Llamaremos

$$\Lambda_n(\lambda) = \log(E(\exp \langle \lambda, Z_n \rangle)) \quad (4.3.26)$$

Hipótesis 4.1. Para todo λ perteneciente a \mathbb{R}^d existe como real extendido el siguiente límite

$$\Lambda(\lambda) = \lim_{n \rightarrow \infty} \frac{\Lambda_n(n\lambda)}{n} \quad (4.3.27)$$

y además el origen pertenece al interior del dominio efectivo de Λ

Definición 4.6. Puntos expuestos de Λ^* . Un punto $y \in \mathbb{R}^d$ es un punto expuesto de Λ^* , si para algún θ perteneciente a \mathbb{R}^d y $\forall x \neq y$ se cumple que

$$\langle \theta, y \rangle - \Lambda^*(y) > \langle \theta, x \rangle - \Lambda^*(x) \quad (4.3.28)$$

θ es llamado el hiperplano expuesto de Λ^* y llamaremos \mathcal{F} al conjunto de puntos expuestos de Λ^* (ver figura 4.4).

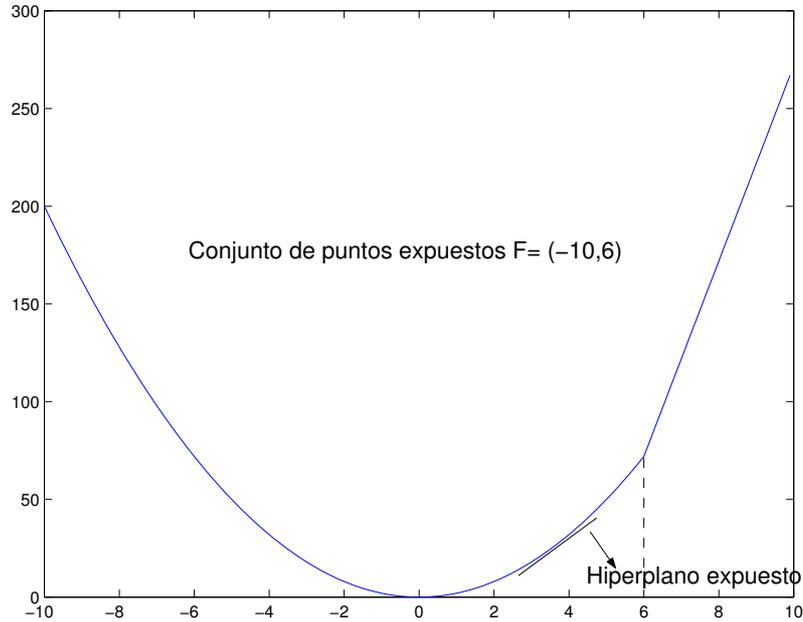


Figura 4.4: Puntos expuestos

Definición 4.7. Una función convexa $\Lambda: \mathbb{R}^d \rightarrow (-\infty, \infty)$ es esencialmente suave si:

1. $D_\Lambda^\circ \neq \emptyset$
2. Λ es diferenciable en el interior de su dominio
3. Λ es steep, 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 λ convergente a un punto de la frontera de dicho dominio efectivo.

Teorema 4.3. Teorema de Gärtner-Ellis. Si se verifica la hipótesis 4.1 entonces se cumple que 1. Para todo conjunto H cerrado y G abierto

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(H) \leq - \inf_{x \in H} \Lambda^*(x) \quad (4.3.29)$$

$$- \inf_{x \in G \cap \mathcal{F}} \Lambda^*(x) \leq \liminf_{n \rightarrow \infty} \frac{1}{n} \log \mu_n(G) \quad (4.3.30)$$

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

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

Idea de la demostración. La demostración de la cota superior es básicamente la misma que en el Teorema de Cramér, es decir demostrarlo para compactos y usando la hipótesis

de que el origen pertenece al interior del dominio efectivo, extenderlo para cerrados. En el caso de la cota inferior, se presenta el problema que no es posible usar directamente la ley de los grandes números, por lo que la demostración utiliza la cota inferior para verificar que la nueva medida tiende a 1 en cualquier abierto centrado en su media.

Observaciones y ejemplos

1. Existen casos donde se verifica un LDP y sin embargo no se cumplen las hipótesis del teorema enunciado. Por ejemplo Dembo y Zeitouini señalan el caso de una sucesión de variables aleatorias exponenciales de parámetro n . En este caso para $x \geq 0$

$$\mathbf{P}(Z_n > x) = \int_x^\infty n \exp(-nz) dz = \exp(-nx) \quad (4.3.31)$$

Es decir que se verifica un LDP con función de velocidad $I(x) = x$. En este caso al calcular la FLGM se obtiene

$$\Lambda_n(\lambda) = \log\left(\frac{n}{n-\lambda}\right) \quad \lambda < n \quad (4.3.32)$$

$$\Lambda_n(\lambda) = \infty \quad \lambda \geq n \quad (4.3.33)$$

Calculando el límite de 4.1, se cumple que $\Lambda(\lambda) = 0$ para $\lambda < 1$ y $\Lambda(\lambda) = \infty$ en otro caso. Por lo tanto $\Lambda^*(x) = x$ para $x \geq 0$ y $\Lambda^*(x) = \infty$ en otro caso. Por lo tanto se que cumple $\Lambda^*(x) = I(x)$. Sin embargo en este caso el conjunto de puntos expuestos contiene solamente al origen y por lo tanto el teorema anterior brinda como cota inferior la trivial, si el conjunto no contiene al origen.

2. Existen otros resultados donde no se puede aplicar el teorema ya que la función Λ no es diferenciable en algún punto del dominio. Un ejemplo es la sucesión Y_n donde $\mathbb{P}(Y_n = n) = 1/2 = \mathbb{P}(Y_n = -n)$ y donde interesa estudiar el gran desvío de la ley de $Z_n = Y_n/n$ En este caso

$$\mathbf{P}(Z_n \in (-1, 1)) = 0 \quad (4.3.34)$$

Calculando $\Lambda_n(n\lambda) = \log(\frac{1}{2}(\exp(n\lambda) + \exp(-n\lambda)))$. Haciendo el límite se llega a que $\Lambda(\lambda) = |\lambda|$. Por lo tanto $\Lambda^*(x) = 0$ para $-1 \leq x \leq 1$ e $\Lambda^*(x) = \infty$ en otro caso. Ahora bien la parte 2 del Teorema 6.5.16 en su cota inferior del LDP conduce a

$$\frac{1}{n} \log \mathbf{P}(Z_n \in (-1, 1)) = -\infty \not\geq -\inf \Lambda^*(x) = 0 \quad (4.3.35)$$

Esta contradicción se debe a que la parte 2 del 6.5.16 no se cumple porque la hipótesis de que la función $\Lambda(\lambda)$ sea diferenciable en el interior de su dominio no se verifica.

3. Fraccional 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 Λ_n y realizando el límite se puede ver que este existe solo si $H \leq 1/2$, es decir siempre que se tenga dependencias de corto rango. Este ejemplo permite ver que si bien no hay una hipótesis explícita sobre la dependencia de los procesos, las condiciones impuestas sobre la función Λ , están relacionadas con la debilidad de su dependencia.

4.4. El régimen asintótico de buffer grande

4.4.1. El análisis de un enlace

Sea $S_n = X(-1) + \dots + X(-n)$, con $S_0 = 0$ como vimos antes el tamaño de la cola en el enlace está dado por

$$Q_0 = \sup_{n \geq 0} S_n \quad (4.4.1)$$

Teorema 4.4. Probabilidad de Pérdida en un enlace *Suponemos que el proceso X es estacionario y ergódico. Sea*

$$\Lambda_n(\lambda) = \log(\mathbf{E}(\exp(\lambda S_n/n))) \quad (4.4.2)$$

Suponemos que el siguiente límite

$$\Lambda(\lambda) = \lim_{n \rightarrow \infty} \frac{1}{n} \log(\mathbf{E}(\exp(\lambda S_n))) \quad (4.4.3)$$

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

$$\limsup_{q \rightarrow \infty} \frac{\log \mathbb{P}(Q_0 \geq q)}{q} \leq -\delta \quad (4.4.4)$$

$$\delta = \sup(\lambda > 0 : \Lambda(\lambda) < 0) \quad (4.4.5)$$

Si la cota inferior del gran desvío

$$\limsup_{n \rightarrow \infty} \frac{\log \mathbb{P}(S_n \geq xn)}{n} \geq \Lambda^*(x) \quad (4.4.6)$$

se verifica para todo $x > 0$, entonces

$$\liminf_{q \rightarrow \infty} \frac{\log \mathbb{P}(Q_0 \geq q)}{q} \geq -\delta \quad (4.4.7)$$

Este Teorema se puede probar de manera simple bajo las hipótesis realizadas obteniendo la cota inferior y superior usando desigualdades clásicas de la Teoría de Probabilidad (ver por ejemplo [9]). En este trabajo veremos su demostración usando el principio de contracción y la noción de 'sample path' LDP ya que es más ilustrativa en el contexto de la tesis. Previamente a introducir los Teoremas referidos a 'sample path LDP' y el principio de contracción veremos un lema que se utilizará en la demostración y la interpretación del Teorema anterior.

Lema 4.1.

$$\delta = \sup(\lambda > 0 : \Lambda(\lambda) < 0) = \inf_{\tau > 0} (\tau \Lambda^*(1/\tau)) \quad (4.4.8)$$

Prueba:

$$\delta = \sup(\lambda > 0 : \Lambda(\lambda) < 0) \quad (4.4.9)$$

$$= \sup(\lambda > 0 : \sup_{x \in \mathbb{R}} [x\lambda - \Lambda^*(x)] < 0) \quad (4.4.10)$$

$$= \sup(\lambda > 0 : x\lambda - \Lambda^*(x) < 0, \forall x) \quad (4.4.11)$$

$$= \sup(\lambda > 0 : \lambda < \Lambda^*(x)/x, \forall x > 0) \quad (4.4.12)$$

$$= \inf_{\tau > 0} (\tau \Lambda^*(1/\tau)) \quad (4.4.13)$$

Consideraciones sobre el Teorema 4.4

- En la figura 4.5 se muestra una interpretación de este Teorema. Recordemos que $X(n) = A(n) - C(n)$, es decir 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 μ 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 4.5 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 consideramos 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: $P(Q > q) \approx \exp(-q\delta)$.
- Analicemos el caso de tráfico gaussiano (μ, σ^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 δ que verifica la ecuación $C = \frac{\Lambda_A(\delta)}{\delta}$ es $\delta = 2(C - \mu)/\sigma^2$. Por lo tanto para una media μ dada, se verifica 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.
- Analicemos el caso de una cola M/M/1. El tráfico que ingresa a la cola es Poisson de parámetro μ_A y el servicio es exponencial de parámetro μ_C . En este caso el punto de operación (ver ejemplo 4.3.22 para el caso poissoniano) queda:

$$\Lambda(\lambda) = \Lambda_A(\lambda) + \Lambda(\lambda)_C = 0 \quad (4.4.14)$$

$$\mu_A(\exp(\lambda) - 1) + \mu_C(\exp(-\lambda) - 1) = 0 \quad (4.4.15)$$

$$\delta = \log\left(\frac{\mu_C}{\mu_A}\right) \quad (4.4.16)$$

$$\mathbf{P}(Q_0 \geq B) \approx \left(\frac{\mu_A}{\mu_C}\right)^B \quad (4.4.17)$$

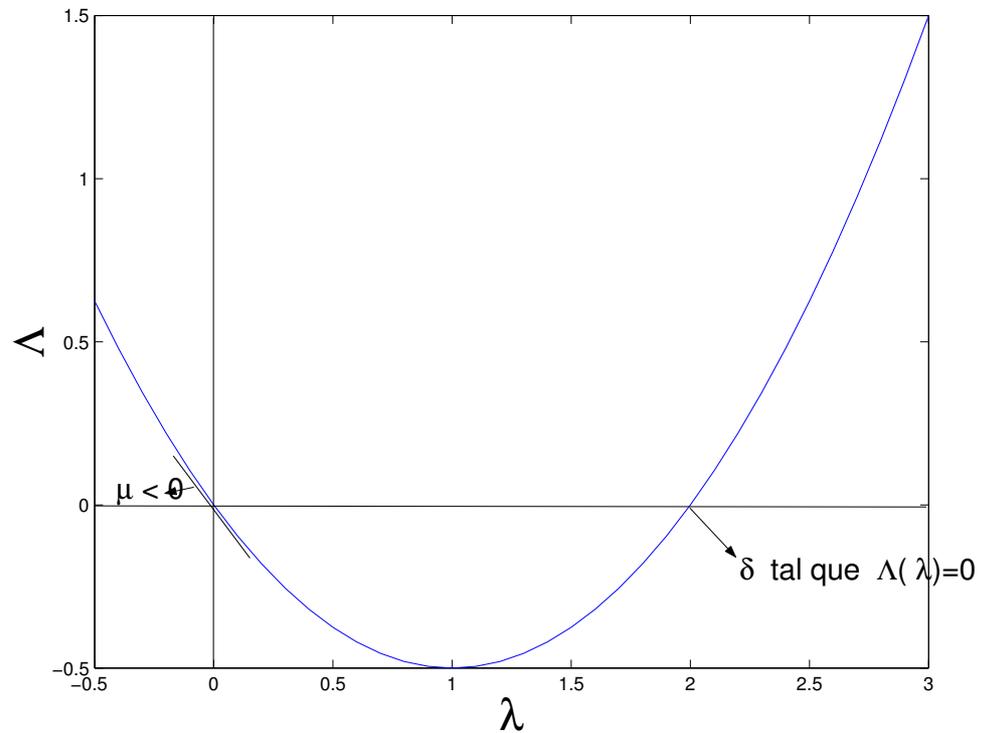


Figura 4.5: Teorema 4.4

4.4.2. Teorema de Mogulskii

Para demostrar el teorema 4.4 se comenzará viendo un teorema clásico de la teoría de Grandes desviaciones que es el teorema de Mogulskii. Luego se verán dos generalizaciones de este teorema. La primera es debida a Dembo y Zajic [17]. En este trabajo se generaliza el resultado de Mogulskii para el caso de variables no i.i.d. El segundo resultado debido a Ganesh y O'Connell [25], permitirá aplicar el principio de contracción para calcular la probabilidad de pérdida de un enlace. Los resultados de Cramer y Garner-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 y exista overflow. Pero para muchos problemas interesará también

saber cómo se llena el buffer, o cual es el tiempo en que es más probable que ocurra un overflow. Para responder a estas preguntas, es necesario estudiar las grandes desviaciones de las trayectorias de los procesos involucrados. El primer resultado que se verá en este sentido es el teorema de Mogulskii. Al igual que en los casos anteriores no haremos la demostración en detalle, sino que realizaremos en este caso un esbozo de las ideas que ayudan a entender el problema y el resultado al que se arriba.

Teorema 4.5. *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 \widehat{S}_n como anteriormente y en este caso el teorema de Cramér dice como se comporta esta sucesión en el límite. Se define ahora la familia de v.a. indexadas por t :*

$$\widehat{S}_n(t) = \frac{1}{n} \sum_{i=1}^{\lfloor nt \rfloor} X_i \quad 0 \leq t \leq 1 \quad (4.4.18)$$

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

$$I(\phi) = \begin{cases} \int_0^1 \Lambda^*(\dot{\phi}(t)) dt, & \text{si } \phi \in \mathcal{AC}, \phi(0) = 0 \\ \infty, & \text{en otro caso} \end{cases} \quad (4.4.19)$$

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

Antes de ver la idea de la demostración se verá una interpretación del Teorema. En la figura 4.6 se muestra la interpretación de $\widehat{S}_n(t)$ para $n = 10$.

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

$$\mathbf{P}(\|\widehat{S}_n - \phi\|_\infty < \varepsilon) \approx \exp(-n \inf_{\|\psi - \phi\|_\infty < \varepsilon} \int_0^1 \Lambda^*(\dot{\psi}(t)) dt) \quad (4.4.20)$$

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

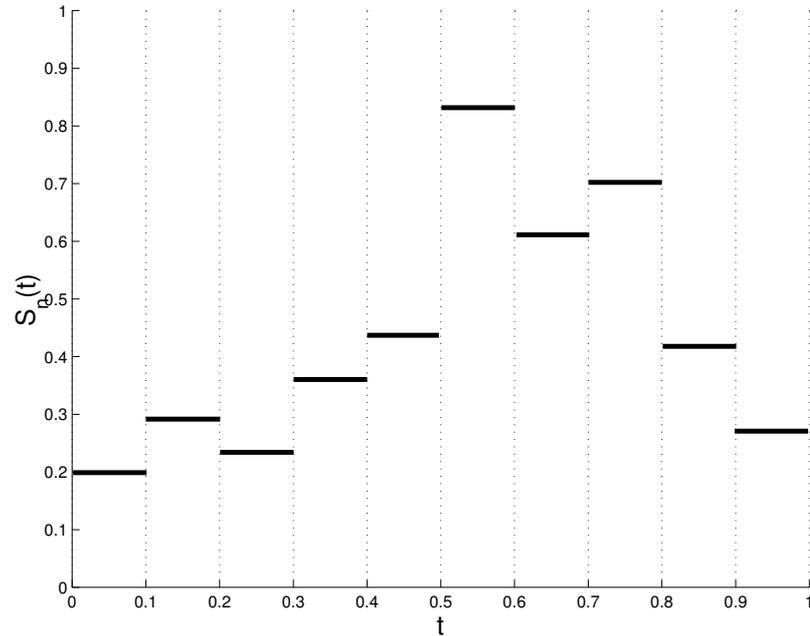


Figura 4.6: $\widehat{S}_n(t)$ para $n = 10$

$$\exp\left(-n \int_0^1 \Lambda^*(\dot{\phi}(t)) dt\right) \quad (4.4.21)$$

La interpretación gráfica se muestra en la figura 4.7.

La parte más ilustrativa de la demostración se puede bosquejar de la siguiente manera. Para cualquier partición de $[0,1]$ con $0 < t_1 < t_2 < \dots < t_j < 1$ definimos la sucesión $Y_n = (\widehat{S}_n(t_1), \widehat{S}_n(t_2) - \widehat{S}_n(t_1), \dots, \widehat{S}_n(t_j) - \widehat{S}_n(t_{j-1}))$. Sea $\lambda = (\lambda_1, \dots, \lambda_j)$, se puede calcular la función Λ a la que tienden las Λ_n de la sucesión Y_n en \mathbb{R}^j . Utilizando además que las X son v.a. i.i.d., operando se llega a que:

$$\frac{1}{n} \lim_{n \rightarrow \infty} \log \mathbf{E} \exp^{\langle \lambda, Y_n \rangle} = \sum_{i=1}^j (t_i - t_{i-1}) \Lambda(\lambda_i) \quad (4.4.22)$$

A partir de esta expresión se puede calcular su transformada de Fenchel-Legendre. Sea

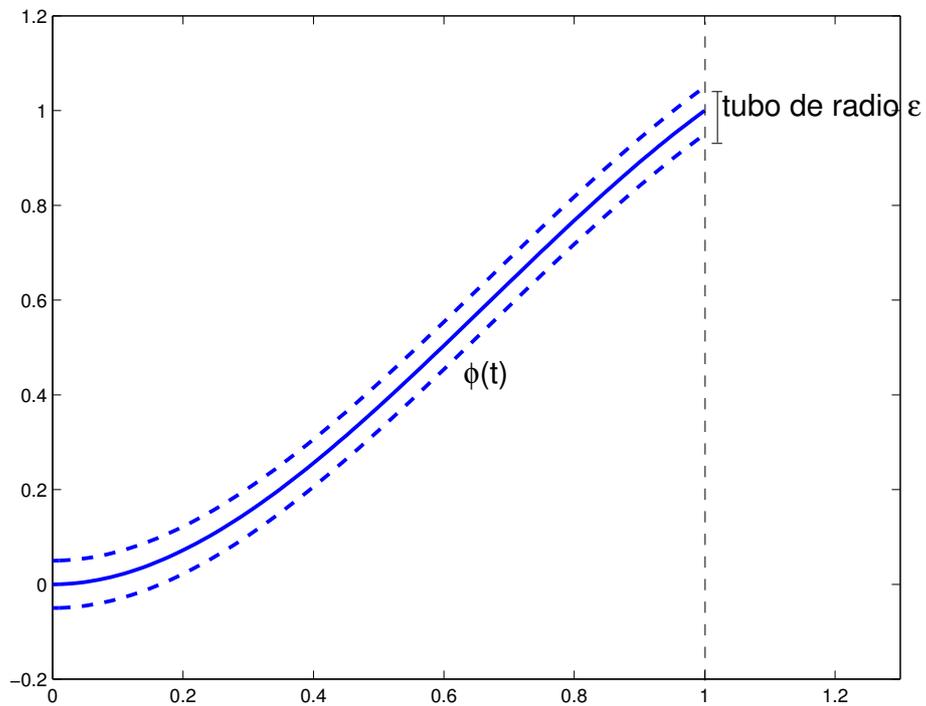


Figura 4.7: Un tubo de radio ε

$$a = (a_1, \dots, a_j)$$

$$\sup_{\lambda} [\langle \lambda, a \rangle - \sum_{i=1}^j (t_i - t_{i-1}) \Lambda(\lambda_i)] = \sum_{i=1}^j (t_i - t_{i-1}) \Lambda^*\left(\frac{a_j}{t_j - t_{j-1}}\right) \quad (4.4.23)$$

De aquí se puede ver de manera aproximada que si hacemos tender $j \rightarrow \infty$ la última sumatoria se transforma en la integral de la tesis del teorema de Mogulskii.

La demostración formal para concluir esto último define una aproximación poligonal $Z_n(t)$ a $\widehat{S}_n(t)$ y demuestra que las medidas de una y otra son exponencialmente equivalentes y utiliza la aproximación poligonal para formalmente demostrar a partir de la expresión 4.4.23 la tesis del teorema.

4.4.3. Teorema de Dembo-Zajic

Este teorema generaliza el teorema anterior para el caso de variables no i.i.d, en particular variables con dependencia débil. Sean las variables aleatorias X_i definidas en un espacio \mathcal{X} . Definimos $\widehat{S}_n(t) = \frac{1}{n} \sum_{i=1}^{\lfloor nt \rfloor} X_i$ con $t \in [0, 1]$

Hipótesis 4.2. Sea $j \in \mathbb{N}$ y $0 = t_0 < t_1 < \dots < t_j \leq 1$, sea $Y_n = (\widehat{S}_n(t_1), \widehat{S}_n(t_2) - \widehat{S}_n(t_1), \dots, \widehat{S}_n(t_j) - \widehat{S}_n(t_{j-1}))$. Entonces Y_n satisface un LDP en \mathcal{X}^j con buena función de velocidad:

$$I_j(y) = \sum_{i=1}^j (t_i - t_{i-1}) I\left(\frac{y_i}{t_i - t_{i-1}}\right) \quad (4.4.24)$$

donde $y = (y_1, \dots, y_m)$ e $I(\cdot)$ es la buena función de velocidad convexa asociada con el LDP de $\{\widehat{S}_n(1)\}$

Teorema 4.6. Dembo y Zajic. Asumiendo que la hipótesis 4.2 es cierta, entonces el proceso $\{\widehat{S}_n(t)\}$, satisface un LDP en el espacio $L^\infty[0, 1]$ con buena función de velocidad convexa

$$I_\infty(\phi) = \int_0^1 \Lambda^*(\dot{\phi}) dt \quad \phi \in \mathcal{AC}, \quad (4.4.25)$$

$$I_\infty(\phi) = \infty \quad \text{en otro caso} \quad (4.4.26)$$

siendo $\Lambda^*(\cdot)$ la buena función de velocidad convexa asociada con el LDP de $\widehat{S}_n(1)$ y \mathcal{AC} , el espacio de las funciones absolutamente continuas en $[0, 1]$, con $\phi(0) = 0$.

En este mismo trabajo Dembo y Zajic prueban que la Hipótesis 4.2 es válida si los procesos son estacionarios y satisfacen ciertas condiciones de mixing. Utilizando el teorema del límite proyectivo de Dawson-Garner (Teorema 5.2) el LDP anterior puede ser extendido de $[0, 1]$ a \mathbb{R}_+ . El problema es que la topología del límite proyectivo (convergencia uniforme sobre compactos) no es lo suficientemente fuerte para las aplicaciones de interés. En particular es fácil ver que la función de distribución de la cola 4.4.1, no es continua en esta topología y por lo tanto no es posible aplicar el principio de contracción para obtener el LDP de la cola. Para tal fin se puede considerar la sucesión de funciones $\phi_n(t) = t/n$ para $0 \leq t \leq n$ y $\phi(t) = 1$ para $t \geq n$. Ahora bien $\phi_n \rightarrow \phi \equiv 0$ uniformemente en conjuntos compactos pero $\sup(\phi_n) = 1$ para todo n , mientras que $\sup(\phi) = 0$.

Por esta razón Ganesh y O'Connell demuestran el teorema que enunciaremos a continuación pero sobre el espacio

$$\mathcal{Y} = \{\phi \in \mathcal{C}(\mathbb{R}_+) : \lim_{t \rightarrow \infty} \frac{\phi(t)}{1+t} \text{ existe}\}, \quad (4.4.27)$$

equipado con la norma:

$$\|\phi\| = \sup_t \left| \frac{\phi(t)}{t+1} \right| \quad (4.4.28)$$

Sobre este espacio sí existe continuidad de la función 4.4.1 y por lo tanto será aplicable el Principio de Contracción para encontrar el LDP del tamaño de cola. Ganesh y O'Connell extienden el resultado de Dembo y Zajic del espacio $L^\infty[0, 1]$ al espacio \mathcal{Y} .

4.4.4. Teorema de Ganesh-O'Connell

Teorema 4.7. *Suponemos que para todo λ el límite 4.3.27 existe como real extendido y el $\widehat{S}_n(t)$ satisface un LDP según 4.4.25 (Dembo-Zajic). Si Λ es diferenciable en el origen entonces $\widehat{S}_n(t)$ satisface un LDP en el espacio \mathcal{Y} (4.4.27) con función buena de velocidad*

$$I_\infty(\phi) = \int_0^\infty \Lambda^*(\dot{\phi}) dt \quad \phi \in \mathcal{AC}_1(\mathbb{R}_+) \cap \mathcal{Y}, \phi(0) = 0 \quad (4.4.29)$$

$$I_\infty(\phi) = \infty \quad \text{en otro caso} \quad (4.4.30)$$

4.4.5. 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.

Teorema 4.8. *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 $y \in \mathcal{X}_2$ definimos

$$J(y) \triangleq \inf\{I(x) : x \in \mathcal{X}_1, y = f(x)\} \quad (4.4.31)$$

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

- Si I controla un LDP asociado con la familia de medidas de probabilidad μ_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.

4.4.6. LDP para el tamaño de la cola de un enlace

A partir de la caracterización del 'sample path' LDP del proceso $\widehat{S}_n(t)$ podemos deducir el resultado respecto al LDP de Q_0 (Teorema 4.4), utilizando el principio de contracción y la desigualdad de Jensen.

Sea $\widehat{S}_n(t) = \frac{1}{n} \sum_{i=1}^{[nt]} X_i$, de donde $Q_0/n = \sup_{t>0}(\widehat{S}_n(t))$, es decir $Q_0/n = f(\widehat{S}_n(\cdot))$. Aplicando el principio de contracción y el teorema anterior obtenemos la siguiente función de velocidad para Q_0 :

$$J(q) = \inf \left\{ \int_0^\infty \Lambda^*(\dot{\phi}) dt : \sup_{t>0} \phi(t) = q \right\} \quad (4.4.32)$$

$$(4.4.33)$$

Luego para todo camino $\phi(t)$ existe un τ donde se alcanza el supremo. El supremo se alcanza porque las funciones pertenecen al espacio \mathcal{Y} donde el $\lim_{t \rightarrow \infty} \phi(t)/t = \mu$ existe y como la cola es estable (es decir que existe una distribución estacionaria del tamaño de la cola) la media del proceso de arribo a la cola debe ser menor que la media del tiempo de servicio, y por lo tanto $\mu < 0$.

Consideremos el camino ψ definido como $\dot{\psi} = q/\tau$ en $(0, \tau]$ y $\dot{\psi} = \mu$ en otro caso. Observando que $\mathbf{E}(\overline{S}_n) = \mu$, se cumple que $\Lambda^*(\mu) = 0$, y aplicando la desigualdad de Jensen ya que Λ^* es convexa se verifica

$$\int_0^\infty \Lambda^*(\dot{\psi}) dt = \int_0^\tau \Lambda^*(\dot{\psi}) dt = \tau \Lambda^*(q/\tau) \leq \int_0^\infty \Lambda^*(\dot{\phi}) dt \quad (4.4.34)$$

y por lo tanto ψ realiza el ínfimo en el espacio considerado. De aquí se puede obtener el

resultado del teorema 4.4

$$J(q) = \inf_{\tau > 0} \tau \Lambda^*(q/\tau) = \delta \cdot q \quad (4.4.35)$$

$$\delta = \sup(\lambda > 0 : \Lambda(\lambda) < 0) = \inf_{\tau > 0} (\tau \Lambda^*(1/\tau)) \quad (4.4.36)$$

La última ecuación se debe al lema 4.1. De esta manera para un enlace y un proceso de entrada y servicio que cumplen con las hipótesis del teorema 4.4, es posible conocer su probabilidad de pérdida cuando el buffer tiende a infinito.

4.4.7. Extensión de los resultados anteriores al análisis de una red

Para extender los resultados anteriores al caso de una red, debemos 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 [33]. En él se establece el siguiente teorema.

Teorema 4.9. *Sea un enlace que recibe d flujos de arribo $\mathbf{X} = (X^1, \dots, X^d)$, tiene un buffer FIFO infinito inicialmente vacío y un servicio estocástico de velocidad C . Sea*

$$\mathbf{A}_n = \sum_1^n \mathbf{X}_i \quad B_n = \sum_1^n C_i \quad \mathbf{S}_n(t) = (\mathbf{A}_{[nt]}/n, B_{[nt]}/n) \quad (4.4.37)$$

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

- (H1). Para todo λ , el $\sup_k \mathbf{E}[\exp^{\lambda(X_k + C_k)}] < \infty$
- (H2) Para todo $\lambda \in \mathbb{R}^d$ el límite

$$\Lambda(\lambda) = \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbf{E}[\exp^{\lambda \cdot S_n(1)}] \quad (4.4.38)$$

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

$$I_\infty(\phi) = \int_0^1 \Lambda^*(\dot{\phi}) dt \quad \phi \in \mathcal{AC}_\dagger \quad (4.4.39)$$

$$I_\infty(\phi) = \infty \quad \text{en otro caso} \quad (4.4.40)$$

donde Λ^* es la transformada de FL de Λ

- (H3) El proceso de arribo y de servicio son asintóticamente independientes en el sentido de que

$$\Lambda^*(\mathbf{x}, c) = \Lambda_a^*(\mathbf{x}) + \Lambda_b^*(c) \quad (4.4.41)$$

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

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

$$\Lambda_d^*(z) = \inf\{\beta\Lambda_a^*(\mathbf{x}/\beta) + \sigma\Lambda_a^*\left(\frac{\mathbf{z} - \mathbf{x}}{\sigma}\right) + \beta\Lambda_b^*(c) + (1 - \beta)\Lambda_b^*\left(\frac{z - x}{1 - \beta}\right)\} \quad (4.4.42)$$

$$\beta, \sigma \in [0, 1], c \in \mathbb{R}, \beta + \sigma \leq 1, x \leq \beta c \quad (4.4.43)$$

Este resultado asume que la cola está inicialmente vacía, se llega a un resultado similar partiendo de la cola en estado estacionario.

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 la performance de cualquier enlace. Sin embargo para que esto sea cierto el tráfico de salida debe verificar las hipótesis (H1),(H2) y (H3) del teorema anterior. En un trabajo reciente Ganesh y o'Connell [24] han encontrado que en general no es cierto que el tráfico de salida verifique la hipótesis (H2). La hipótesis H2 habitualmente es referida como la propiedad de "geodésica lineal". Esto se debe a la siguiente propiedad: En que se cumpla la hipótesis H2 la trayectoria más probable que conduce al gran desvío $\overline{S}_n \geq a$ es una trayectoria lineal. Observemos que $\overline{S}_n = \widehat{S}_n(1)$ y por lo tanto

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}(S_n \geq a) = \lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P}(S_n = \overline{S}_n(1) \geq a) \leq - \inf_{\phi(1) \geq a} \int_0^1 \Lambda^*(\phi'(t)) dt \quad (4.4.44)$$

Cómo Λ^* es convexa y $\phi(0) = 0$ aplicamos la desigualdad de Jensen y obtenemos que

$$\int_0^1 \Lambda^*(\phi'(t)) dt \geq \Lambda^*(\phi(1)) \quad (4.4.45)$$

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

$$- \inf_{\phi(1) \geq a} \int_0^1 \Lambda^*(\phi'(t)) dt \geq \Lambda^*(a) \quad (4.4.46)$$

pero para $\phi(t) = a.t$ se verifica que $\int_0^1 \Lambda^*(\phi'(t))dt = \Lambda^*(a)$ 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 como vimos de que la función de velocidad pueda ser expresada como la integral de una función convexa (H2). En el trabajo de Ganesh y O'Connell se muestran contraejemplos 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 H2. Si bien para $d = 1$ esta propiedad no se conserva en general, si lo hace si el servicio del enlace es determinístico. Este último es el caso estudiado por Chang [9], para el análisis de un tipo particular de redes "intree networks". Para el caso en que se tienen diversas clases de tráfico a la entrada del enlace $d > 1$, se muestra con un contraejemplo en dicho trabajo, que aún en el caso de servicio determinístico, la propiedad de geodésica lineal no necesariamente se preserva al atravesar un enlace. Ahora bien, del trabajo del Dembo y Zajic referido antes, se obtiene que si se cumplen hipótesis de mixing entonces se satisface la hipótesis H2 ('geodésica lineal'), el problema se da por la dependencia que se introduce al atravesar una cola y que lleva a que pueda dejar de cumplirse esta hipótesis. Esta propiedad de geodésica lineal es característica de la asintótica de buffer grande pero como se verá en el capítulo siguiente en el caso de muchas fuentes las trayectorias más probables al overflow en ese régimen no son necesariamente lineales.

Capítulo 5

Grandes desviaciones aplicadas al análisis de performance en el régimen asintótico de muchas fuentes

5.1. Introducción

En la aplicación de la teoría de grandes desviaciones al análisis de redes de Telecomunicaciones se ha trabajado sobre dos asintóticas: la asintótica de buffer grande y la asintótica de muchas fuentes. En el capítulo anterior se han reseñado los principales resultados del régimen de buffer grande. En este se estudia el comportamiento en el caso en que el enlace está alimentado por un número infinito de fuentes (escalando el buffer y la capacidad del enlace con el crecimiento de las fuentes).

5.2. El régimen asintótico de muchas fuentes

5.2.1. Introducción

Consideremos 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 como se muestra en la figura 7.1

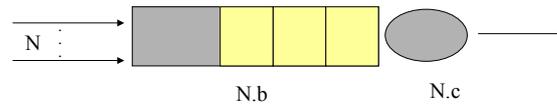


Figura 5.1: Regimen de muchas fuentes

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

5.2.2. Ancho de Banda Efectivo

Utilizaremos la definición de ancho de banda efectivo (EB por su sigla en inglés) desarrollada por Kelly[28]. Sea $X[0, t]$ la cantidad de trabajo acumulado que arriba desde una fuente en el intervalo $[0, t]$. Asumimos que el proceso $X[0, t]$ tiene incrementos estacionarios. Se define el ancho de banda efectivo como:

$$\alpha(\lambda, t) = \frac{1}{\lambda t} \log(E(e^{\lambda X[0, t]})) \quad (5.2.1)$$

Surge la pregunta de porqué el nombre de **ancho de banda efectivo**. La idea es que el valor de esta función para un cierto punto de operación (λ^*, 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 del sistema. 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 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 $f(\alpha_A(\lambda, t), \alpha_B(\lambda, t))$ a los efectos de tener formas simples de agregación.

Estas condiciones son verificadas por la función $\alpha(\lambda, t)$ ver por ejemplo [9]. En particular el EB de la agregación de dos fuentes independientes es la suma de los EB de c/u de ellas. Pueden existir otras funciones '**ancho de banda efectivo**' que cumplan estas condiciones, pero en ciertos contextos esta definición tiene una interpretación muy natural. Como se vio para el régimen de buffer grande, si se supone que el servidor es determinístico de capacidad c , la generatriz de momentos que utilizada en el teorema 4.4 para obtener la probabilidad de pérdida en un enlace será

$$\Lambda(\lambda) = \lim_{t \rightarrow \infty} \lambda \alpha(\lambda, t) - c\lambda \quad (5.2.2)$$

y en ese caso

$$\lim_{q \rightarrow \infty} \frac{\log \mathbb{P}(Q_0 \geq q)}{q} \leq -\delta \text{ siendo} \quad (5.2.3)$$

$$\delta = \sup(\lambda > 0 : \Lambda(\lambda) \leq 0) \quad (5.2.4)$$

$$\text{es decir la solución de :} \quad (5.2.5)$$

$$\alpha(\lambda, \infty) = c \quad (5.2.6)$$

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 λ^* 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.

5.2.3. Probabilidad de pérdida en un enlace

A partir de la expresión del ancho de banda efectivo, Wischik [44] demostró para el régimen asintótico de muchas fuentes, usando la Teoría de las grandes desviaciones, una expresión que permite calcular la probabilidad de pérdida de un enlace. Esta expresión ya había sido encontrada por otros autores previamente [12], pero la demostración de Wischik utilizando 'sample path LDP', permitirá obtener algunas conclusiones para el tipo de redes que interesa analizar. Sea un enlace donde confluyen N fuentes de K tipos diferentes y donde las fuentes son i.i.d.. Asignamos una capacidad c y un tamaño de buffer b por fuente. Sea $\alpha_j(\lambda, t)$ el EB de una fuente del tipo j -ésimo que alimenta el enlace. Sea $C = Nc$, $B = Nb$ y ρ_j la proporción de fuentes de tipo j . Se verá que:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log(P(Q > B)) = -I \quad (5.2.7)$$

$$I = \inf_t \sup_{\lambda} ((c + bt)\lambda - \lambda t \sum_j^K \rho_j \alpha_j(\lambda, t)) \quad (5.2.8)$$

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

$$P(Q > B) \approx \exp^{\inf_t \sup_{\lambda} ((C+Bt)\lambda - \lambda t \sum_j^K N \rho_j \alpha_j(\lambda, t))} \quad (5.2.9)$$

es decir,

$$P(Q > B) \approx \exp^{-NI} \quad (5.2.10)$$

La forma de calcular la probabilidad de pérdida en un enlace es resolver la doble optimización 5.2.8, de donde se obtiene un punto de operación (λ^*, 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 overflow del enlace. El parámetro λ se denomina parámetro espacial y λ^* indica el grado de multiplexado estadístico del enlace.

Para llegar a este resultado se seguirán los principales pasos de la demostración de Wischik.

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} lo notaremos como $\mathbf{x}(0, \infty)$ y cuando se trunca al conjunto $s, s + 1, \dots, t$ lo notaremos $\mathbf{x}(s, t)$ para $s < t$. Notaremos $\mathbf{1}$ al proceso que toma valores 1 constantes en todos los instantes de tiempo. Trabajaremos 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 sample path LDP para \mathbf{X}^N y luego aplicar el principio de contracción para obtener un LDP para la probabilidad de pérdida. El primer paso será encontrar un LDP para los procesos truncados.

Hipótesis 5.1. *Sea la función logarítmica generadora de momentos Λ_t^N para $\lambda \in \mathbb{R}^t$ definida por*

$$\Lambda_t^N(\lambda) = \frac{1}{N} \log \mathbf{E}(\exp(N\lambda \cdot \mathbf{X}^N(0, t])) \quad (5.2.11)$$

Se asume que para cada t y λ la función límite generadora de momento

$$\Lambda_t(\lambda) = \lim_{N \rightarrow \infty} \Lambda_t^N(\lambda) \quad (5.2.12)$$

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

Teorema 5.1. *Bajo la hipótesis anterior el vector $\mathbf{X}^N[0, t]$, para todo t fijo, satisface un LDP en \mathbb{R}^t con buena función de velocidad*

$$\Lambda_t^*(\mathbf{x}(0, t]) = \sup_{\lambda \in \mathbb{R}^t} \lambda \cdot \mathbf{x}(0, t] - \Lambda_t(\lambda) \quad (5.2.13)$$

como consecuencia directa de aplicar el teorema de Gärtner-Ellis en \mathbb{R}^t .

El siguiente resultado extiende el anterior a procesos en $(0, \infty)$ y surge como aplicación directa del teorema de Dawson-Gartner. Este teorema permitirá extender un LDP de un

'espacio reducido' hacia un 'espacio más grande'. Se utilizará a continuación para extender un LDP sobre una colección finita de v.a. a un LDP sobre una sucesión infinita de v.a.. Veremos algunas definiciones previas.

Definición 5.1. *Sea J un conjunto parcialmente ordenado. Un sistema proyectivo $(\mathcal{Y}_j, p_{ij})_{i \leq j \in J}$ consiste en un conjunto de espacios topológicos Hausdorff $\mathcal{Y}_{j \in J}$ y mapeos continuos $p_{ij} : \mathcal{Y}_i \rightarrow \mathcal{Y}_j$, tal que $p_{ik} = p_{ij} \circ p_{jk}$ cuando $i \leq j \leq k$. El límite proyectivo del sistema es $\mathcal{X} = \lim_{\leftarrow} \mathcal{Y}_j$, que es el subconjunto del espacio topológico producto $\prod_{j \in J} \mathcal{Y}_j$ consistente de todos los elementos $\mathbf{x} = (y_j)_{j \in J}$ para los cuales $y_i = p_{ij}(y_j)$ cuando $i \leq j$ equipada con la topología inducida por \mathcal{Y}*

Teorema 5.2. Teorema de Dawson – Gärrner

Sea μ_n una familia de medidas de probabilidad en \mathcal{X} tal que para cualquier $j \in J$, las medidas de probabilidad de $\mu_n \circ p_j^{-1}$ sobre \mathcal{Y}_j satisfacen un LDP con buena función de velocidad $I_j(\cdot)$. Entonces μ_n satisface un LDP con buena función de velocidad

$$I(x) = \sup_{j \in J} \{I_j(p_j(x))\}, \quad x \in \mathcal{X} \quad (5.2.14)$$

Se puede extender con este teorema el resultado obtenido para $\mathbf{X}^N[0, t]$ a $\mathbf{X}^N[0, \infty]$. En este caso $\mathcal{Y}_j = \mathbb{R}^j$ y $(p_{ij})_{i \leq j \in J} : \mathbb{R}^i \rightarrow \mathbb{R}^j$ y $\mathcal{X} = \lim_{\leftarrow} \mathcal{Y}_j$, siendo \mathcal{X} el espacio de los procesos a valores reales indexados por t . La proyección p_j trunca el proceso en \mathcal{X} a sus valores en $(0, j]$.

Teorema 5.3. *Bajo la Hipótesis 5.1, $\mathbf{X}^N[0, \infty]$ satisface un LDP en \mathcal{X} con la topología del límite proyectivo con buena función de velocidad*

$$I(\mathbf{x}) = \sup_t \Lambda_t^*(\mathbf{x}(0, t]) \quad (5.2.15)$$

Este resultado tal como está formulado no podrá ser aplicado directamente para analizar la probabilidad de overflow, ya que la ecuación de Loynes 4.2.7 no es continua en la topología de convergencia uniforme sobre compactos, y por lo tanto no se puede aplicar el principio de contracción directamente. El siguiente ejemplo muestra que dicha función no es continua. Se considera una cola de capacidad constante C a la que arriba tráfico según: $x_t^N = C$ para $0 < t < N$, $x_t^N = C + 1$ para $t = N$ y $x_t^N = 0$ para $t > N$. Esta sucesión converge al proceso constante \mathbf{C} uniformemente sobre cualquier compacto. Sin embargo, $Q(x^N) = \sup_{t > 0} (x(0, t] - Ct) \rightarrow 1 \neq Q(\mathbf{C}) = 0$. Para poder aplicar al resultado 5.2.15 el principio

de contracción usando la función de Loynes, es necesaria encontrar una topología donde esta función sea continua y extender el resultado 5.2.15 en esta topología. Por esta razón Wischik trabaja con la topología definida sobre el espacio \mathcal{X} por la norma:

$$\|x\| = \sup_{t>0} \left| \frac{x(0, t]}{t} \right| \quad (5.2.16)$$

Para extender el LDP de una topología a la otra es necesario usar el siguiente Teorema (ver demostración en[18]), también llamado Inverso del Principio de contracción

Teorema 5.4. *El Inverso del principio de Contracción.* Sean \mathcal{X} e \mathcal{Y} espacios topológicos de Hausdorff. Sea $g : \mathcal{Y} \rightarrow \mathcal{X}$ una biyección continua y sea ν_n una familia de medidas de probabilidad con decaimiento exponencial sobre \mathcal{Y} . Si $\{\nu_n \diamond g^{-1}\}$ satisface un LDP con función de velocidad I , entonces ν_n , satisface el LDP con función buena de velocidad $J \triangleq I(g(\cdot))$.

En la demostración de este teorema no es posible usar como en la demostración del principio de contracción que al ser continua la función entonces los abiertos y cerrados del codominio se transforman en abiertos y cerrados del dominio, porque la función g va de $\mathcal{Y} \rightarrow \mathcal{X}$. Es necesario entonces una condición adicional. Esta condición es la de decaimiento exponencial (exponentially tight). Usando esta hipótesis se prueba la cota superior de un LDP débil sobre compactos y luego se lo extiende a un LDP fuerte al ser la medida de decaimiento exponencial. Para la cota inferior sobre abiertos, la hipótesis de decaimiento exponencial se utiliza para asegurar que en la transformación no se 'acumule medida en el infinito'.

Para aplicar el inverso del principio de contracción y extender estos resultados con la transformación identidad que lleva de $(\mathcal{X}, \|\cdot\|) \rightarrow (\mathcal{X}, p)$ (p es la topología del límite proyectivo), es necesario probar que la medida de probabilidad de \mathbf{X}^N en el espacio $(\mathcal{X}, \|\cdot\|)$ es de decaimiento exponencial. La idea de esta topología es que controla lo que sucede en escalas grandes de tiempo. Para que \mathbf{X}^N sea de decaimiento exponencial es necesario agregar alguna hipótesis sobre el comportamiento en 'escala de tiempos grandes' de \mathbf{X}^N . Wischik agrega la siguiente hipótesis:

Hipótesis 5.2. 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 definimos la función:

$$\Lambda_t^N(\lambda) = \frac{1}{v(t)} \Lambda_t^N(\mathbf{1}\lambda v(t)/t) \quad (5.2.17)$$

para $\lambda \in \mathbb{R}$. Por la hipótesis 5.1, existe una vecindad abierta del origen en la cual el límite

$$\Lambda_t(\lambda) = \lim_{N \rightarrow \infty} \Lambda_t^N(\lambda) \quad (5.2.18)$$

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

$$\Lambda(\lambda) = \lim_{t \rightarrow \infty} \Lambda_t(\lambda) \quad (5.2.19)$$

existen uniformemente en λ . Asumiremos que para λ en alguna vecindad abierta del origen, el límite

$$\sqrt{\frac{v(t)}{\log t}} (\Lambda_t^N(\lambda) - \Lambda(\lambda)) \rightarrow 0 \quad (5.2.20)$$

uniformemente en λ cuando $t, N \rightarrow \infty$

Con la hipótesis 5.2 Wischik prueba que la medida de probabilidad de \mathbf{X}^N en el espacio $(\mathcal{X}, \|\cdot\|)$ tiene decaimiento exponencial y por lo tanto se verifica el siguiente teorema aplicando el Inverso del principio de contracción.

Teorema 5.5. (*Sample-path LDP para procesos promediados*) Suponemos que \mathbf{X}^N verifica las hipótesis 5.1 y 5.2. Entonces, el proceso satisface un LDP en el espacio $(\mathcal{X}, \|\cdot\|)$ con buena función de velocidad \mathbf{I} definida en 5.2.15

Una vez encontrado un LDP para el proceso de entrada al enlace es posible aplicar el principio de contracción con la función de Loynes y encontrar el LDP para el tamaño del buffer en equilibrio. Para aplicar el principio de contracción la función debe ser continua y en este caso es necesario que la cola sea estable en el sentido de Loynes. La función Q_0 será continua solo si la cola es estable, es decir si la media del proceso de entrada es menor que la media del proceso de servicio. La función del tamaño de la cola no es continua en todo el espacio $(\mathcal{X}, \|\cdot\|)$, sino que solo sobre aquellos procesos de entrada que aseguren que la cola tiene una distribución estacionaria. Para eso Wischik define el espacio $(\mathcal{X}_\mu, \|\cdot\|)$ con

$$\mathcal{X}_\mu = \left\{ x \in \mathcal{X} : \frac{x(0, t]}{t} \leq \mu \text{ a partir de un cierto } t_0 \right\} \quad (5.2.21)$$

donde μ es mayor que la velocidad media de \mathbf{X}^N .

Wischik demuestra que el LDP se mantiene si se restringe el espacio a $(\mathcal{X}_\mu, \|\cdot\|)$. Posteriormente para obtener el LDP de la cola se utilizará el espacio $(\mathcal{X}_C, \|\cdot\|)$, siendo C la capacidad del enlace que será mayor que la media de la velocidad del tráfico de entrada.

Teorema 5.6. (*Sample-path LDP para procesos promediados*) Suponemos que \mathbf{X}^N verifica las hipótesis 5.1 y 5.2. Entonces, el proceso satisface un LDP en el espacio $(\mathcal{X}_\mu, \|\cdot\|)$ con buena función de velocidad \mathbf{I} definida en 5.2.15

Este teorema se prueba viendo dos cosas:

$$\mathbf{P}(\mathbf{X}^N \in \mathcal{X}_\mu) = 1 \quad (5.2.22)$$

y que $\{\mathbf{x} : I(\mathbf{x}) < \infty\} \subset \mathbf{X}^N$. Al demostrar los dos enunciados anteriores, aplicando el Lema 4.1.5 del libro de Dembo y Zeituni [18], se ve que la función de velocidad es la misma en el espacio restringido.

A partir del Teorema anterior es posible obtener los siguientes teoremas que son el principal resultado de esta sección, aplicando el principio de contracción

Teorema 5.7. (*fórmula infsup*) Suponemos que \mathbf{X}^N verifica las hipótesis 5.1 y 5.2 y que $\Lambda'_t(\lambda \mathbf{1}) < Ct$ en $\lambda = 0$ para todo t , entonces $Q(\mathbf{X}^N)$ satisface un LDP con buena función de velocidad

$$I(b) = \inf_{\mathbf{x} \in \mathcal{X}_C: Q(\mathbf{x})=b} I(\mathbf{x}) \quad (5.2.23)$$

$$= \inf_{t>0} \inf_{\mathbf{x} \in \mathbb{R}^t: \mathbf{x}(0, t]=b+Ct} \Lambda_t^*(\mathbf{x}(0, t]) \quad (5.2.24)$$

$$= \inf_t \sup_\lambda (\lambda(b + Ct) - \Lambda_t(\lambda \mathbf{1})) \quad (5.2.25)$$

Teorema 5.8. (*fórmula samplepath*) 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 λ^* es alcanzado entonces

$$\mathbf{x}^*(0, t^*] = \nabla \Lambda_{t^*}(\lambda^* \mathbf{1}) \quad (5.2.26)$$

Antes de comentar la demostración interesa ver la importancia de estos resultados. El resultado del Teorema 5.7 permite estimar la probabilidad de pérdida en un enlace por

$$P(Q > B) \approx \exp^{-N.I} \quad (5.2.27)$$

Es decir que si es posible conocer o estimar para el tráfico de entrada $\Lambda_t(\lambda \mathbf{1})$, se tendrá una aproximación de la probabilidad de pérdida en el enlace. Por otra parte el teorema 5.8 dice como será en este caso la "trayectoria más probable" que conduce al overflow. En el caso de la asintótica de "buffer grande" se vio que esta trayectoria era lineal, sin embargo la expresión 5.2.26, permite verificar para casos simples que en el caso de la asintótica de muchas fuentes el camino más probable $\nabla \Lambda_{t^*}(\lambda^* \mathbf{1})$ 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 λ^* indica el grado de multiplexado estadístico. Si el valor de λ^* 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 $\lambda^* \rightarrow \infty$, se estará reservando un EB próximo al valor de pico y por lo tanto aprovechando poco el multiplexado estadístico.

Para probar el teorema anterior, se prueba primero que la función Q_0 es continua en $(\mathcal{X}_\mu, \|\cdot\|)$ con $\mu < C$, lo cual se hace considerando una sucesión $x^k \rightarrow x$ en este espacio y verificando que $|Q(x^k) - Q(x)| \rightarrow 0$. La ecuación 5.2.23 es consecuencia directa de la aplicación del principio de contracción. La única observación es que el espacio es \mathcal{X}_C pero puede usarse \mathcal{X}_μ con μ mayor que la media del proceso y menor que C , ya que de la ecuación 5.2.22 se puede ver que $\mathbf{P}(\mathbf{X}^N \in \mathcal{X}_C / \mathcal{X}_\mu) = 1$. El resto de la demostración se basa en el principio de contracción por un lado, y en el hecho de que la función I y la función Λ^* son buenas funciones de velocidad. Debido a esto, el "sample path" \mathbf{x}^* es efectivamente alcanzado. Además se ve que en \mathcal{X}_μ con $\mu < C$ y mayor que la media del proceso, el tiempo t^* también es efectivamente alcanzado ya que de lo contrario existiría una sucesión

de tiempos t_n tal que $\mathbf{x}(0, t_n]/t_n \rightarrow C$, lo que no puede pasar en \mathcal{X}_μ .

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 :

$$I = \sup_{\lambda} \lambda C - \Lambda_{\mathbf{1}}(\lambda \mathbf{1}) \quad (5.2.28)$$

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 5.2.28. Este resultado se utilizará en la sección siguiente. Eun y Shroff [22] 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 5.7 y 5.8 se verifican para el caso de un enlace con buffer B finito.

5.2.4. 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 punta a punta. Se trata de analizar la performance de toda la red de punta a punta, no solo 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 [43] que en el régimen asintótico de muchas fuentes, la función ancho de banda efectivo $\alpha(\lambda, t)$ 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. Veremos 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 Λ_t^N para la entrada agregada es

$$\Lambda_t^N(\lambda) = \log \mathbf{E}(\exp(\lambda \cdot \mathbf{X}^{(N)})) \quad (5.2.29)$$

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

$$\tilde{\Lambda}_t^N(\lambda) = \log \mathbf{E}(\exp(\lambda \cdot \tilde{\mathbf{X}}^{(N)})) \quad (5.2.30)$$

Teorema 5.9. Caracterización de la salida en tiempo finito *Asumimos que $\mathbf{X}^{(N)}$ satisface las hipótesis 5.1 y 5.2, y que dicho proceso es estacionario y con media estrictamente menor que C . Llamando a la función límite generadora de momentos de la entrada $\lim_{N \rightarrow \infty} \log \mathbf{E}(\lambda \cdot \mathbf{X}^{(N)}(0, t]) = \Lambda_t(\lambda)$, entonces el proceso de salida $\tilde{\mathbf{X}}^{(N)}$ satisface la hipótesis 5.1 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 $\exp(\lambda \cdot \tilde{\mathbf{X}}^{(N)})$ es uniformemente integrable. De estas dos conclusiones se ve que $\mathbf{E}(\exp(\lambda \tilde{X}^{(N)}(0, t])) - \mathbf{E}(\exp(\lambda X^{(N)}(0, t])) \rightarrow 0$

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 5.7 para lo cual la salida debe verificar la hipótesis 5.2. 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 (llamaremos wq) dada por la métrica

$$d(\mathbf{x}, \mathbf{y}) = |Q(\mathbf{x}) - Q(\mathbf{y})| + \sum_{t=1}^{\infty} \frac{1 \wedge |x_t - y_t|}{2^t} \quad (5.2.31)$$

Teorema 5.10. Caracterización de la salida en tiempo largos *Asumimos que $\mathbf{X}^{(N)}$ satisface las hipótesis 5.1 y 5.2, 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}_μ, wq) (μ 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}_μ, wq) .

5.3. Ajuste de la asintótica

Likhanov y Mazumdar en su trabajo [32] encontraron una aproximación que ajusta la asintótica de la probabilidad de pérdida en el régimen asintótico de 'muchas fuentes' mejorando su aproximación a las pérdidas reales en un enlace con un buffer finito. La base para el cálculo de esta asintótica mejorada es el teorema de Bahandur-Rao ([18]). Esta asintótica mejorada para valores grandes de N se puede aproximar por [11]:

$$P(\text{perdidas}) \approx \exp^{-N.I - \frac{1}{2} \log(4.\pi.N.I)} \quad (5.3.1)$$

Esta aproximación la usaremos más adelante en la propuesta de ingeniería de tráfico.

Capítulo 6

Análisis de performance en una red MPLS con buffers pequeños

6.1. Introducción

Al finalizar el capítulo 5 se vio un resultado encontrado por Wischick para el análisis de performance de redes. En dicho trabajo se demuestra 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 pero no se adapta a una red MPLS.

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. También se vio que la ocupación del buffer tiende a cero con N tendiendo a infinito.

Interesa para la tesis analizar si es posible estudiar la performance de una red MPLS estudiando la performance de una red 'ficticia' equivalente. Esta red ficticia es idéntica a la red real excepto porque a cada enlace interior, el tráfico de los LSPs llega tal como se presenta antes de ingresar a la red. En esta 'red ficticia' los enlaces anteriores no afectan el tráfico que arriba a un enlace interior. La pregunta que surge es si en algún caso este análisis

simplificado con la red ficticia brinda los mismos resultados que sobre la red real. También interesa estudiar bajo qué condiciones el análisis con la red ficticia es al menos una buena aproximación o al menos cuándo dicho análisis no subestima los parámetros de QoS de la red. Este estudio tiene interés porque el análisis con la red 'ficticia' es obviamente más simple que con la red real más compleja. Si se logra encontrar estas condiciones, se contará con una herramienta de análisis más simple aunque aproximada pero que permitirá trabajar en línea.

Las respuestas a estas preguntas se derivan de un trabajo reciente (Mayo de 2003) de Mazumdar, Ozturk y Likhanov [34] que se analizará en este capítulo. A partir del citado trabajo en la tesis, al final del presente capítulo, se realizará un estudio que permite encontrar las condiciones mencionadas.

En el trabajo de Mazumdar 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. Los resultados de este trabajo permitirán un análisis de performance de los parámetros de calidad de servicio de una red MPLS con buffers pequeños. Es de hacer notar, que esta situación no corresponde formalmente a la asintótica de muchas fuentes ya que en este caso la capacidad se escala con N pero no el tamaño del buffer.

6.2. Análisis de performance de una red alimentada por muchos flujos y buffer pequeño

A continuación se resumen los principales resultados del trabajo de Mazumdar, Ozturk y Likhanov [34].

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 asumimos 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 notamos $X_t^{m,N}$ 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.

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

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

Asumiremos que $\mu_m^N \rightarrow_{N \rightarrow \infty} \mu_m$. Asumiremos que $X^{m,N}(0,t)/N$ satisface un LDP con función buena de velocidad $I_t^{X^m}(x)$:

$$-\inf_{x \in \Gamma^o} I_t^{X^m}(x) \leq \liminf_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}(X^{m,N}(0,t)/N \in \Gamma) \quad (6.2.1)$$

$$\leq \limsup_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}(X^{m,N}(0,t)/N \in \Gamma) \leq -\inf_{x \in \bar{\Gamma}} I_t^{X^m}(x) \quad (6.2.2)$$

donde $\Gamma \in \mathbb{R}$ es un conjunto de Borel con interior Γ^o y clausura $\bar{\Gamma}$ e $I_t^{X^m}(x) : \mathbb{R} \rightarrow [0, \infty)$ es un mapeo continuo con conjuntos de nivel compactos. Es de hacer notar que 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 6.2.1 vale por Cramer, 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, incluso modelos con dependencias largas: 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

$$\sum_{m \in \mathcal{M}_k} \mu_m < C_k \quad (6.2.3)$$

es decir que todas las colas son estables.

Teorema 6.1. *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 tráficos externos de entrada a la red tal que:*

$$X_{k,0}^{m,N}/N = g_k^m(X_0^{1,N}/N, \dots, X_0^{M,N}/N) + o(1) \quad (6.2.4)$$

bajo la hipótesis de estabilidad 6.2.3, la probabilidad de overflow del buffer viene dada por:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}(\text{overflow en el nodo } k) = -\mathbf{I}_k = \quad (6.2.5)$$

$$- \inf \left\{ \sum_{m=1}^M I_1^{X^m}(x_m) : x = (x_m) \in \mathbb{R}^M, \sum_{m=1}^M g_k^m(x) > C_k \right\} \quad (6.2.6)$$

Veremos 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 , analicemos un enlace j cualquiera de la red y llamemos $X_{j,0}^{m,N}$ ($Y_{j,0}^{m,N}$) al tráfico instantáneo de tipo m entrante (saliente) al enlace j . Se define para $n = 1, \dots, M$ la función

$$f_n(x_1, \dots, x_M, y) = \frac{x_n y}{\max(\sum_{i=1}^M x_i, y)} \quad (6.2.7)$$

Como el buffer es FIFO, si no hay otras hipótesis sobre diferenciación entre las clases, la capacidad del enlace es llenada 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:

$$Y_{j,0}^{n,N} \geq f_n(X_{j,0}^{1,N}, \dots, X_{j,0}^{M,N}, NC_j - B_j(N)) \quad (6.2.8)$$

$$\geq f_n(X_{j,0}^{1,N}, \dots, X_{j,0}^{M,N}, NC_j) - B_j(N) \quad (6.2.9)$$

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

$$Y_{j,0}^{n,N} \leq f_n(X_{j,0}^{1,N}, \dots, X_{j,0}^{M,N}, NC_j) + B_j(N) \quad (6.2.10)$$

ya que el peor caso es que el buffer estuviera lleno solo 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:

$$Y_{j,0}^{n,N}/N = f_n(X_{j,0}^{1,N}/N, \dots, X_{j,0}^{M,N}/N, C_j) + o(1) \quad (6.2.11)$$

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 MPLS que interesan en esta tesis son feed-forward. Mazumdar 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 deduciremos 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.

Primero que nada 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 función buena de velocidad :

$$I_1^{X^m}(y) = -\inf\left\{\sum_{n=1}^M I_1^{X^n}(x_n) : x = (x_n) \in \mathbb{R}^M, g_k^m(x_1, \dots, x_M) = y\right\} \quad (6.2.12)$$

Observar que entre clases hay independencia.

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

$$I_1^{Z_k}(y) = -\inf\left\{\sum_{n=1}^M I_1^{X^n}(x_n) : x = (x_n) \in \mathbb{R}^M, g_k(x_1, \dots, x_M) = y\right\} \quad (6.2.13)$$

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

El resultado 6.2.13 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 solo 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 Mazumdar 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 definimos $\mathbf{L}^{m,N}$ 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

$$\mathbf{L}^{m,N} = \sum_{k \in r^m} k \in r^m \frac{L_k^{m,N}}{\mathbb{E}[X_t^{m,N}]} \quad (6.2.14)$$

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

$$L_k^{m,N} = \mathbb{E}[(X_{k,t}^{m,N} + Q_{k,t-1}^{m,N} - C_{k,t}^{m,N})^+ - Q_{k,t}^{m,N}] \quad (6.2.15)$$

A partir de esta expresión operando se demuestra que

Teorema 6.2.

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbf{L}^{m,N} = - \min_{k \in r^m} \mathbf{I}_k \quad (6.2.16)$$

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. Un punto importante a resaltar y sobre el que volveremos más adelante es que el ratio de pérdidas así definido es una métrica cóncava.

El último resultado de este trabajo que utilizaremos 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.

Asumiremos 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:

$$\mathcal{D} = \{(n_m), m = 1, \dots, M : \lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbf{L}^{m,N} < -\gamma_m\} \quad (6.2.17)$$

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

Teorema 6.3. *Sea \mathcal{D} la región de aceptación para (n_m) definida antes. Se considera el sistema ficticio 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,

$$\overline{\mathcal{D}} \subseteq \mathcal{D} \quad (6.2.18)$$

y si para todo m $\gamma_m = \gamma$, entonces

$$\overline{\mathcal{D}} = \mathcal{D} \quad (6.2.19)$$

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 en lo que resta del capítulo condiciones bajo las cuales el análisis sobre la red ficticia es exacto o una buena aproximación. Este estudio permitirá también entender cuando el análisis sobre la red ficticia no es exacto de qué depende la magnitud del error.

6.3. Aplicación de los resultados anteriores al análisis de performance de una red MPLS

En el caso de una red MPLS (con buffers pequeños) la aplicación de los resultados anteriores es casi directa. Si consideramos los flujos del tipo de tráfico m como el conjunto de flujos que conforman un LSP, a partir del teorema 6.2, podemos estimar la probabilidad de pérdida en cualquier nodo de la red. Para esto si modelamos o medimos trazas del tráfico que ingresa a cada LSP, la función de velocidad de cada LSP vendrá dada por

$$I_1^{X^m}(x) = \sup_{\lambda} \lambda x - \log \mathbb{E} \exp(\lambda \widetilde{X}_0^m) \quad (6.3.1)$$

$$= \sup_{\lambda} \lambda x - \lambda \alpha_m(\lambda, 1) \quad (6.3.2)$$

donde \widetilde{X}_0^m es el valor de tráfico instantáneo de una de las N fuentes i.i.d que atraviesan el LSP. Si un LSP tuviera más de un tipo de fuente, se tendría una combinación de los anchos de banda efectivo de las diferentes fuentes. Conocido o estimado el ancho de banda efectivo $\alpha_m(\lambda, 1)$ de cada LSP, es posible por lo tanto calcular, la probabilidad de pérdida en cualquier enlace de la red MPLS a partir de la ecuación:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}(\text{overflow en el enlace } k) = -\mathbf{I}_k = \quad (6.3.3)$$

$$- \inf \left\{ \sum_{m=1}^M I_1^{X^m}(x_m) : x = (x_m) \in \mathbb{R}^M, \sum_{m=1}^M g_k^m(x) > C_k \right\} \quad (6.3.4)$$

donde la función g_k se puede obtener iterativamente para cada nodo a partir de la función f_n vista antes que depende de las capacidades de los enlaces y de la topología de LSPs de la red. Esto puede fuera de línea resolverse numéricamente y estimar tanto la probabilidad de pérdida en cada enlace como el ratio de pérdidas de punta a punta de cada LSP.

6.4. Otros parámetros de calidad de servicio en estas redes

Los otros parámetros de interés en cuanto a la calidad de servicio, son el retardo y el Jitter. Ahora bien en las redes que estamos considerando con capacidades grandes y buffers pequeños, es importante notar que acotar el retardo y el jitter es equivalente a acotar la probabilidad de pérdidas, ya que los retardos serán muy pequeños siempre que el paquete no se pierda y por lo tanto si se tiene una baja probabilidad de pérdidas se estará asegurando un retardo máximo por enlace pequeño (equivalente al tamaño del buffer (pequeño), sobre su capacidad (grande)). El retardo de punta a punta podremos controlarlo verificando que el retardo máximo en el camino (sumando los pequeños retardos máximos en cada enlace) no supere una cota para esa clase de tráfico.

6.5. ¿Es posible trabajar con la red 'ficticia' en lugar de la real ?

El objetivo que perseguimos en este capítulo es utilizar estas herramientas para realizar ingeniería de tráfico en línea en una red MPLS. La base será el trabajo reseñado en este capítulo de Mazumdar et al. pero es necesario simplificar el cálculo. Se desea evitar aplicar y recalcular las funciones $g_k(x)$ en línea en la red. Como se mencionó antes Mazumdar prueba

que en la red ficticia la región de admisión 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 y si fuera posible hacerlo. Este problema si bien interesa minimizarlo, es menos grave que aceptar conexiones de más y que se dejen de cumplir los requerimientos de QoS. En esta simplificación conociendo los LSPs que atraviesan cada enlace y una estimación del ancho de banda efectivo (de lo que hablaremos en el capítulo siguiente), es posible conocer el ancho de banda efectivo actual que atraviesa cada enlace de la red. Una vez conocido el ancho de banda efectivo total que atraviesa un enlace, la función de velocidad de las pérdidas en ese enlace k se calcula como:

$$I_1^k(x) = \sup_{\lambda > 0} \lambda x - \lambda \sum_{X_m \in k} \alpha_{X_m}(\lambda, 1) \quad (6.5.1)$$

Mazumdar et al. muestran también que si todas las clases de tráfico que usan la red tienen el mismo requerimiento de QoS (en función del ratio de pérdidas), las regiones de aceptación coinciden. Si esto se cumple, sabemos que usando el sistema ficticio en lugar del real no se sobreestimarán la región de aceptación. En el caso general, no siempre es cierto que los requerimientos de QoS de todas las clases sean las mismas. Por lo tanto analizaremos a continuación, en qué condiciones esta aproximación brinda regiones de aceptación iguales al caso real y cual es la magnitud del error cuando no lo es. Este problema se analizará en tres etapas. Primero se estudiarán simulaciones que brindarán ideas respecto de como abordar este punto. Luego, se realiza un análisis teórico para un caso simple con dos nodos. Por último, se verá el resultado principal de este capítulo para un caso más general en una red MPLS. Este punto ha sido estudiado recientemente por R. Casellas [8]. Casellas se concentra en el análisis del caso de la figura 6.1, es decir en una red con dos nodos donde al segundo nodo solo llega un agregado del tráfico del primer nodo. En esta tesis se obtendrá una generalización de ese resultado.

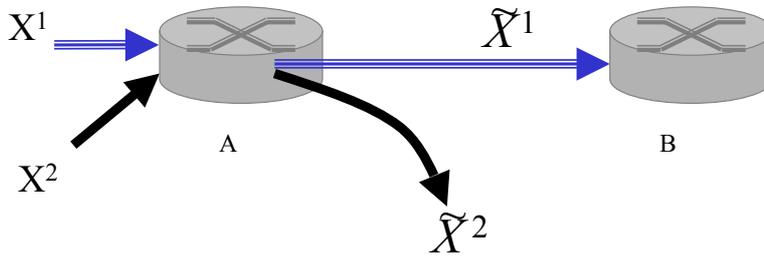


Figura 6.1: Red con dos nodos

6.5.1. Simulaciones

El Teorema 5.10 demostrado por Wischik, dice que si a se agregan N flujos **independientes** e idénticamente distribuidos a un flujo de la salida de un enlace $\tilde{\mathbf{X}}^{(N)}$ entonces dicho agregado verifica el mismo LDP que el agregado de la entrada y se podría aplicar este LDP a la entrada de otro enlace y repetir el procedimiento. Sin embargo para que esto último sea válido, la estructura de la red debería ser como la de la figura 6.2, es decir una red feed-forward y donde cada flujo de la salida $\tilde{\mathbf{X}}^{(N)}$ de un nodo se lo hace llegar a otro nodo siguiente y se lo agrega con otras $N - 1$ copias **independientes** provenientes de salidas de otros $N - 1$ enlaces idénticos. El interés en esta tesis es aplicarlo en un esquema diferente. El caso más simple de interés es por ejemplo el de una red con dos nodos como el de la figura 6.1, es decir donde un agregado de los que atraviesan el primer enlace llega al segundo.

En este caso las hipótesis no son válidas, ya que para cualquier N finito no se tiene como entrada al segundo enlace copias independientes de $\tilde{\mathbf{X}}^{(N)}$ y por lo tanto no se cumplen las hipótesis requeridas para volver a aplicar las mismas herramientas al segundo enlace.

Se analizará a continuación cómo se comporta el segundo enlace cuando a la entrada arriba el agregado de flujos que atravesó el primer enlace. Para analizar este punto se realizaron las siguientes simulaciones:

1. Simular el esquema de la figura 6.1, pero en el caso más simple que es cuando todo el flujo que sale del primer nodo se dirige al segundo. Compararemos el ancho de banda

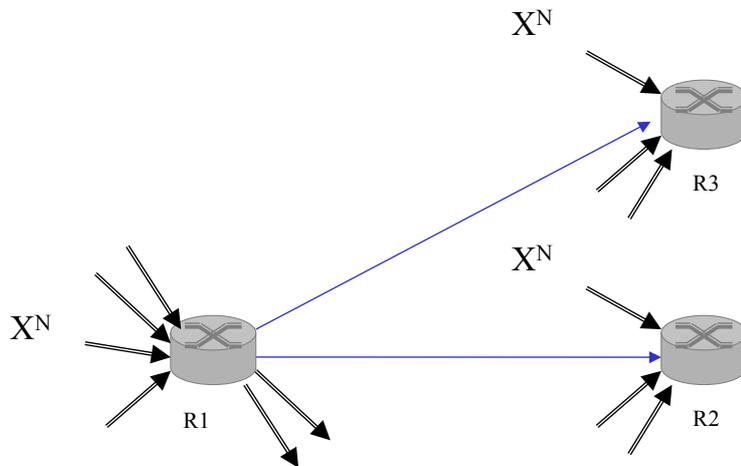


Figura 6.2: Red Intree

efectivo de la entrada y la salida para distintos valores del número de fuentes, del tamaño de buffer y de la capacidad.

2. Simular el esquema de dicha figura (en el caso más simple mencionado) y estudiar la probabilidad de pérdida del segundo enlace en dos condiciones:

- a. Cuando el agregado de flujo de entrada al segundo enlace es la salida del primer enlace
- b. Cuando la entrada al segundo enlace es el agregado de flujos que ingresa al primer enlace directamente.

Para estas simulaciones utilizaremos un modelo de tráfico markoviano ON-OFF, es decir un modelo que corresponde a una cadena de Markov en tiempo continuo con dos estados. En el estado OFF no se emite tráfico y en el estado ON se transmite a una velocidad h_1 . Para un modelo markoviano más general con K estados y velocidades de transmisión h_i en cada estado, el ancho de banda efectivo es [26]:

$$\alpha(s, t) = \frac{1}{st} \log \{ \vec{\pi} \exp(Q + Hs)t \vec{1} \} \quad (6.5.2)$$

donde Q es la matriz generadora infinitesimal de la cadena, $\vec{\pi}$ es la distribución estacionaria de la cadena y H es una matriz diagonal con elementos h_i en la diagonal. Se

utilizó un simulador en tiempo discreto, donde cada enlace está regido por la ecuación de Lyndley. Las fuentes usadas simulan el tráfico markoviano referido antes, simulando la cadena de Markov de dos estados. Se simularon las topologías referidas en la figura 6.3. Por un lado en la Topología A el conjunto de flujos de salida del enlace 1 alimenta el enlace 2 y en la Topología B el flujo de entrada al enlace 2 es el de ingreso al enlace 1 de la Topología A. En la figura 6.4 se compara la probabilidad de pérdida del enlace 2 obtenida mediante simulación en la Topología A y en la Topología B. La gráfica muestra el porcentaje de error cometido al calcular las pérdidas en la red ficticia en lugar de hacerlo en la real. Como se aprecia al aumentar N (cantidad de fuentes) el error relativo disminuye y en este caso, cuando se multiplexan aproximadamente 30 fuentes el error es razonablemente pequeño.

Lo que en realidad interesa es como se comporta el ancho de banda efectivo y la función que debemos optimizar para calcular las pérdidas (que llamaremos $\gamma(s, t)$) a la entrada y a la salida del enlace 1.

$$\gamma(s, t) = (ct + b)s - st\alpha(s, t) \quad (6.5.3)$$

Para este estudio se obtuvieron trazas de tráfico markoviano simulado. Se midieron las trazas de tráfico al ingresar al enlace 1 (X_1) y al salir del enlace 1 (X_2). Con estas dos trazas se estimó el ancho de banda efectivo de ambas trazas con el siguiente estimador. Dada una traza de tráfico de tamaño n se define

$$\widetilde{X}_1 = \sum_1^t x(i) \quad (6.5.4)$$

$$\widetilde{X}_2 = \sum_t^{2t} x(i) \quad (6.5.5)$$

$$\dots\dots\dots \quad (6.5.6)$$

El ancho de banda efectivo de esta traza puede ser estimado por [11]:

$$\alpha_n(s, t) = \frac{1}{st} \log\left(\frac{1}{\lfloor n/t \rfloor} \sum_{i=1}^{\lfloor n/t \rfloor} \exp^{s\widetilde{X}_i}\right) \quad (6.5.7)$$

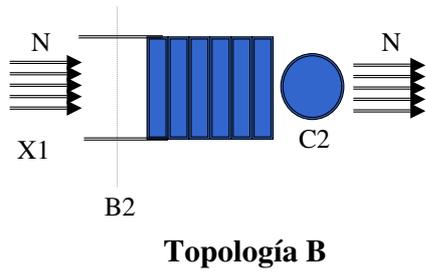
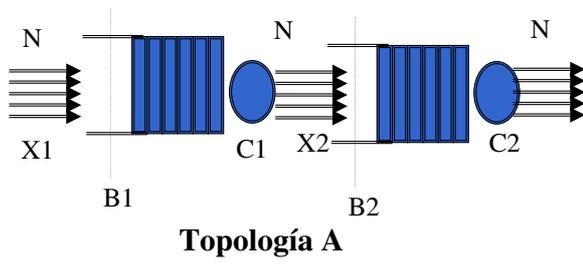


Figura 6.3: Topologías de Simulación

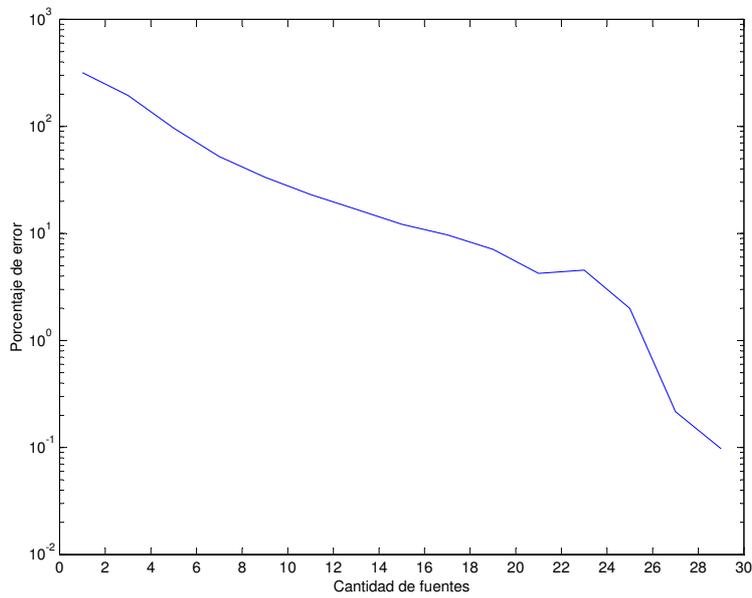


Figura 6.4: Error al calcular las pérdidas en el enlace 2 con la Topología B en lugar de la Topología A

donde $\lfloor c \rfloor$ denota el mayor entero menor o igual que c . En las figuras 6.5 y 6.6 se muestra el ancho de banda efectivo obtenido para $N = 1$ y $N = 20$ respectivamente

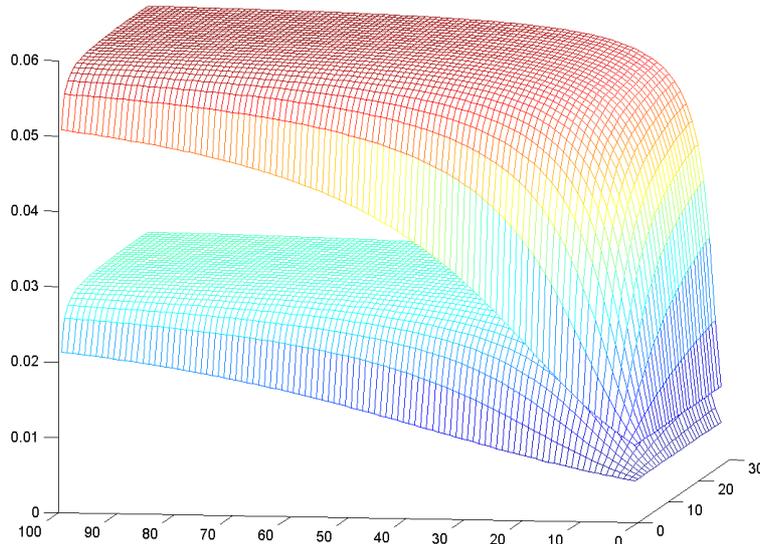


Figura 6.5: Ancho de banda efectivo a la entrada y a la salida del enlace, para $N=1$

Como se puede ver en las figuras el ancho de banda efectivo de la entrada y la salida del enlace son cada vez más similares al aumentar N . Es importante notar que ambas funciones ancho de banda efectivo se acercan más en los valores pequeños de s , que en los valores grandes de s . La observación anterior se debe a que cuando $s \rightarrow 0$, el ancho de banda efectivo tiende al valor medio del proceso y cuando $s \rightarrow \infty$, tiende al valor máximo del proceso. Por lo tanto al aumentar la cantidad de fuentes el proceso de entrada tiene mayor cantidad de arribos con valores en el entorno de la media, que pasan por el enlace sin ser alterados, y los valores muy grandes (donde deben coincidir muchas fuentes emitiendo a su valor máximo), son los que no pasan por enlace pero son valores cada vez menos probables. Es evidente que ambos anchos de banda no pueden coincidir para todo s ya que el ancho de banda de la salida está acotado por la capacidad del primer enlace, mientras que la entrada está acotada por el valor de pico de la fuente, por lo cual al menos por encima de la capacidad del primer

enlace ambas funciones no pueden coincidir. Estas conclusiones también se pueden apreciar en las figuras 6.7, 6.8 y 6.9 donde se muestra la función $\gamma(s, t)$ (logaritmo de la probabilidad de overflow del segundo enlace) para las trazas anteriores con diferentes valores de N y valores de capacidad y tamaño de buffer del segundo enlace fijos.

De estas simulaciones se puede observar que para valores pequeños de s , y valores grandes de N , el ancho de banda efectivo de la salida puede ser aproximado por el de la entrada al enlace con una aproximación razonable. La pregunta es qué implica sobre la red la condición 'valores pequeños de s '. Es importante tener en cuenta que lo que interesa de la función ancho de banda efectivo, es tener una buena aproximación donde se da el punto de operación (punto que resuelve la optimización $\inf_t \sup_s (\gamma(s, t))$). Observamos que para calcular γ , para cada t fijo se resuelve el máximo entre una función convexa ($st\alpha(s, t)$) y una recta $(ct + b)s$, como se aprecia en la figura 6.10.

En estas simulaciones observamos que para obtener una buena aproximación (quizás exacta para $N \rightarrow \infty$), la recta $((ct + b)s)$ debe tener poca pendiente. La pendiente depende de la capacidad del enlace $C2$ y del tamaño del buffer $B2$. La pendiente en el origen de $st\alpha(s, t)$ es el valor medio del tráfico (μ). En una red para tener un buen aprovechamiento de recursos, se buscará siempre que la utilización ($\rho = \mu/C$) sea alta. Habitualmente se trabajará con valores de $C2$ cercanos a μ y por lo tanto si $B2$ es pequeño el óptimo se dará para valores pequeños de s donde la aproximación es buena. La variación con $B2$ del punto óptimo se puede apreciar también en la figura 6.11. Estas simulaciones han sido realizadas en un escenario donde no hay interferencia entre distintos agregados de tráfico, por lo cual no podemos sacar conclusiones generales. Sin embargo, estas simulaciones han permitido observar en un caso simple, algunos aspectos del problema que conducirán a resultados más generales de las secciones siguientes.

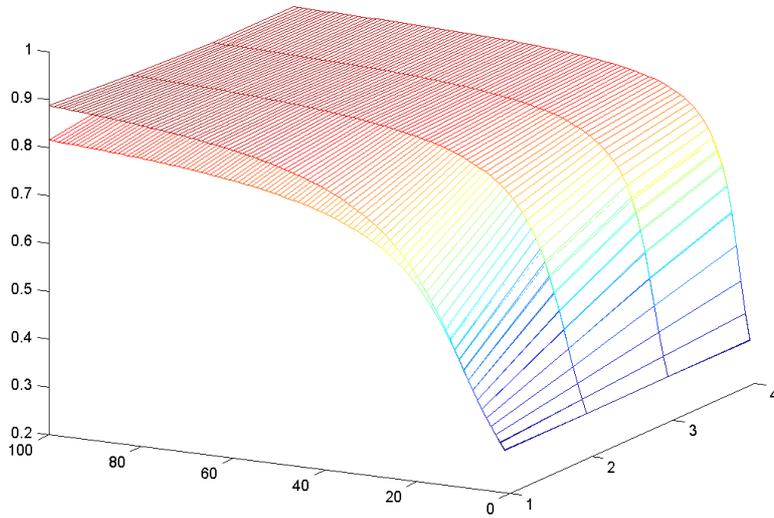


Figura 6.6: Ancho de banda efectivo a la entrada y a la salida del enlace, para $N=20$

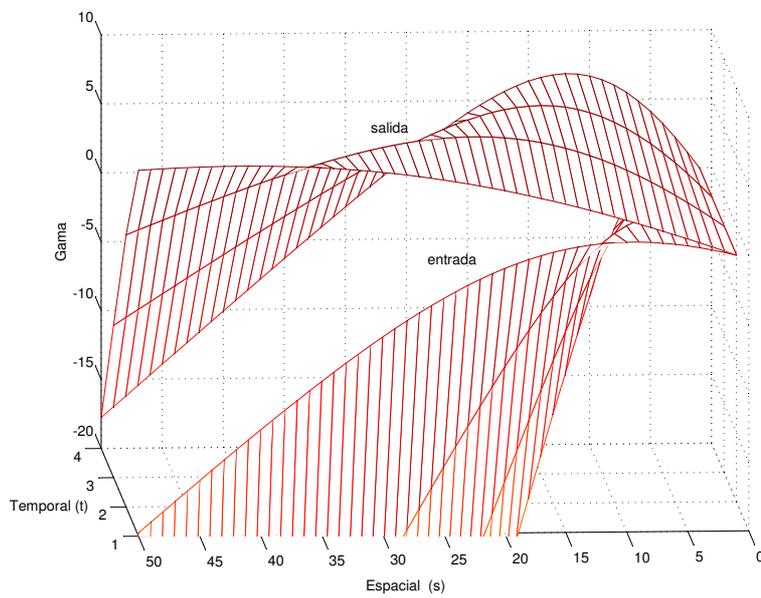


Figura 6.7: $\gamma(s, t)$ para $N=1$

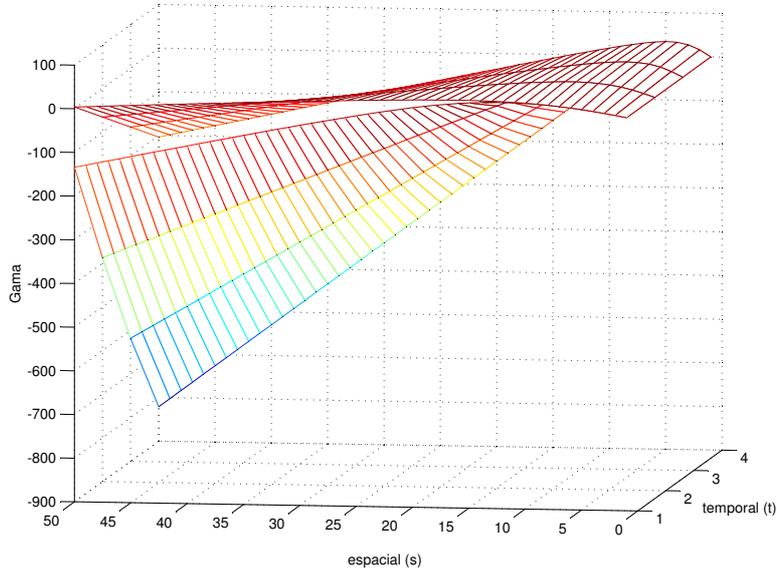


Figura 6.8: $\gamma(s, t)$ para $N=15$

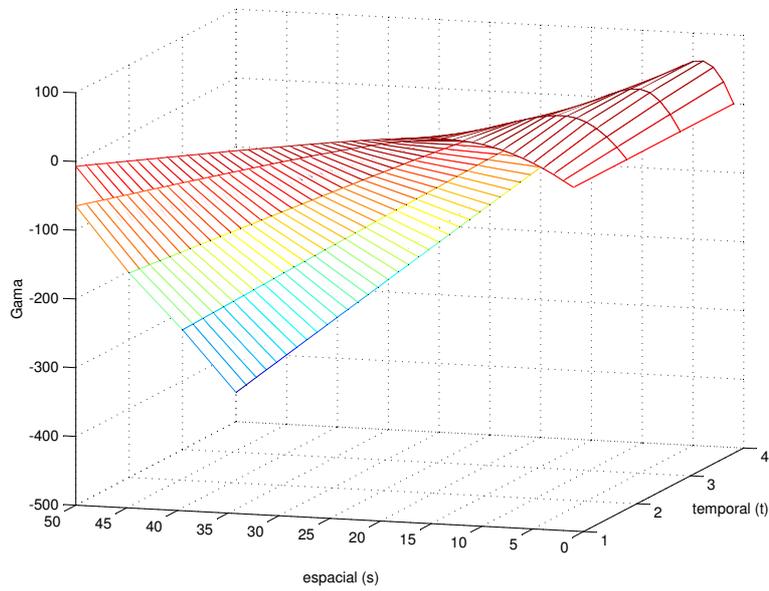


Figura 6.9: $\gamma(s, t)$ para $N=29$

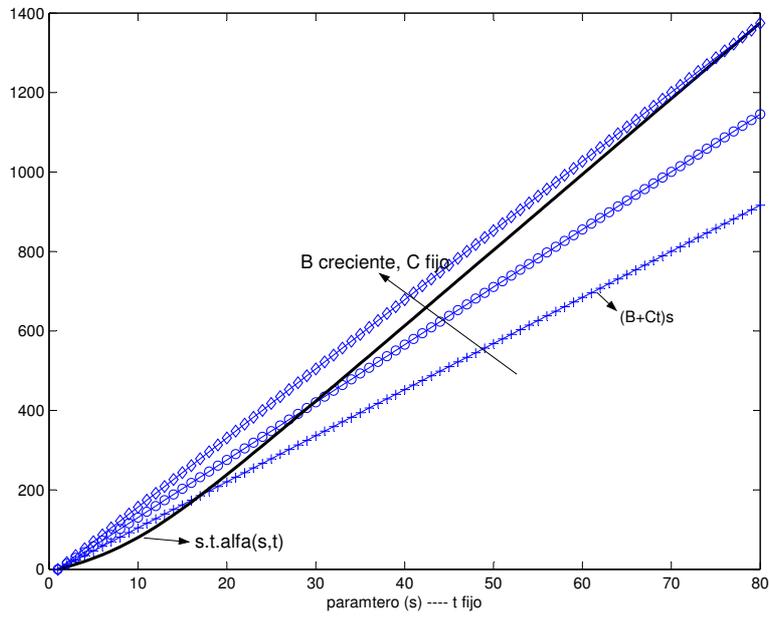


Figura 6.10: Calculo del s óptimo

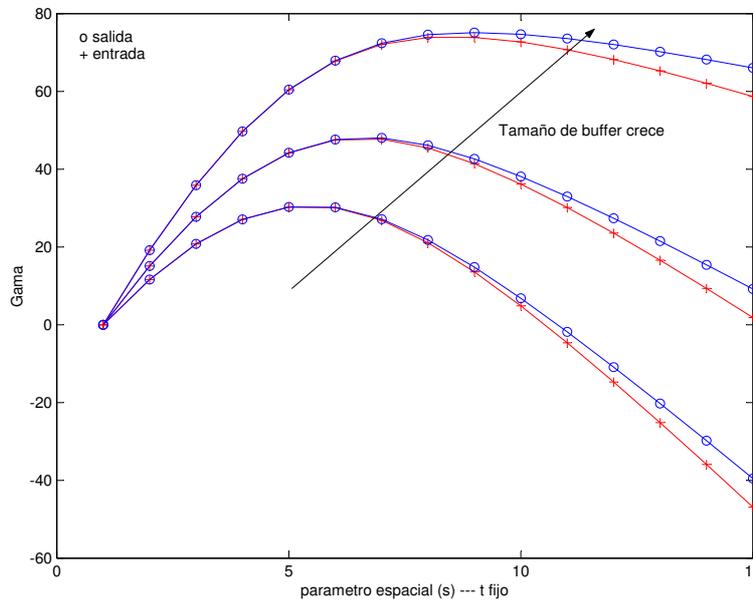


Figura 6.11: Gama(s,t) para N=29,t fijo, C fijo, s y B variables

6.5.2. El análisis de una red con dos nodos

Consideremos primero la red simple de la figura 6.1. Calculemos para el enlace B , la función velocidad de la probabilidad de pérdida en la red ficticia y en la red real. En la red ficticia la función velocidad de grandes desvíos de la probabilidad de pérdida del enlace B es:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}(\text{perdidas en el enlace } B) = -\mathbf{I}_B = \quad (6.5.8)$$

$$-\inf\{I_1^{X^1}(x) : x > C_B\} = I_1^{X^1}(C_B) \quad (6.5.9)$$

Usando la red real, según el trabajo de Mazumdar se tendría para este enlace:

$$\mathbf{I}_B = -\inf\{I_1^{X^1}(x_1) + I_1^{X^2}(x_2) : \frac{x_1 C_A}{\max(C_A, x_1 + x_2)} > C_B\} \quad (6.5.10)$$

En la figura 6.12 se puede ver la región de optimización 6.5.10 gráficamente. En ella se observa el punto (λ, μ) siendo λ la media de la velocidad instantanea del tráfico X^1 y μ la del tráfico X^2 .

Sabemos que $I_1^{X^1}(\lambda) = 0$ e $I_1^{X^2}(\mu) = 0$, por lo tanto la función velocidad en ese punto $I(x) = I_1^{X^1}(\lambda) + I_1^{X^2}(\mu) = 0$. Sabemos también que tanto $I_1^{X^1}$ como $I_1^{X^2}$ son funciones convexas. En la figura 6.12 se muestra la región H que verifica: $\frac{x_1 C_A}{\max(C_A, x_1 + x_2)} > C_B$, que es donde se debe buscar el ínfimo de la función $I(x)$. Por las propiedades vistas de la función velocidad de grandes desvíos, si el punto (λ, μ) , se encuentra dentro de la región H , entonces estamos ante un caso trivial, donde la probabilidad de pérdidas será 1, porque no se cumplen las condiciones de estabilidad de las colas de los enlaces A y B . Si H no incluye el punto (λ, μ) , entonces, por las propiedades de $I(x)$ y de H , el ínfimo se encuentra sobre la frontera de H . Si estuviera en un punto (u, v) del interior de H , entonces si se desplaza el punto por el interior de H sobre la recta $y = v$ y disminuyendo x la función velocidad disminuye. Esto se debe a que $I_1^{X^2}(v)$ no varía y $I_1^{X^1}(x)$ disminuye si x disminuye. Como (λ, μ) no pertenece

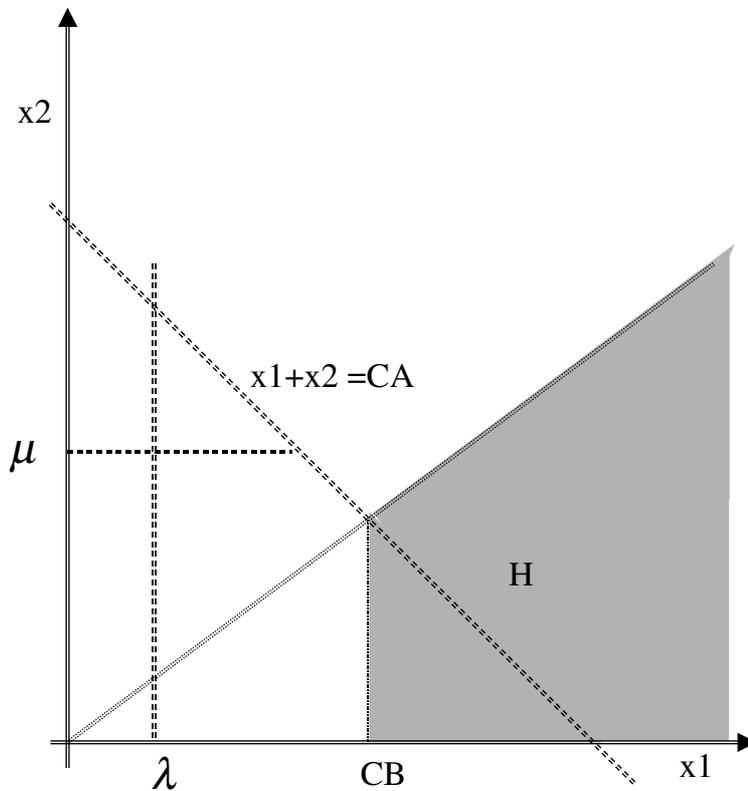


Figura 6.12: Análisis con dos nodos

al interior de H llegamos a la frontera con un valor menor de función velocidad que en el punto interior.

Para estudiar donde se ubica el mínimo sobre la frontera y cuando coincide la solución en la red ficticia y la real, se deben distinguir dos casos:

- $C_B \leq C_A - \mu$, en este caso el mínimo se encuentra sobre la frontera $x_1 = C_B$ y cómo en esta frontera $I_1^{X^1}(x_1)$ es constante, e $I_1^{X^2}(x_2) \geq 0$ es convexa y con $I_1^{X^2}(\mu) = 0$, el ínfimo se da en el punto (C_B, μ) y su valor es $I_1^{X^1}(C_B)$, es decir que la velocidad de grandes desvíos coincide con la del sistema ficticio de la ecuación 6.5.8 y no hay error en aproximar el sistema real por el ficticio.

- $C_B > C_A - \mu$ en este caso el ínfimo se obtendrá sobre la frontera $x_2 = (C_A - C_B)x_1/C_B$, ya que si estuviera sobre la frontera $x_1 = C_B$, con $x_2 < (C_A - C_B)$, haciendo crecer x_2 , disminuye la función velocidad y por lo tanto el mínimo estaría en $x_1 = C_B$, $x_2 = C_A - C_B$ que pertenece a $x_2 = (C_A - C_B)x_1/C_B$. Como se aprecia en este caso la función de velocidad del sistema real y el ficticio no coinciden. Lo que sucede es que el sistema ficticio sobreestima la probabilidad de pérdida y por lo tanto la región de aceptación será menor. Por otra parte se puede ver que el error cometido en este caso está acotado por $I_1^{X^2}(C_A - C_B)$. En un caso real, la velocidad de transmisión es siempre positiva y por lo tanto $I_1^{X^2}(x) = \infty, x < 0$ y en el origen puede tomar un valor finito como en el ejemplo 2 de la sección 4.3.1 o puede tender a infinito como en el caso exponencial (ejemplo 4 de la sección 4.3.1). Esto se ve intuitivamente porque si en el primer enlace se recorta el tráfico a la capacidad C_A , las pérdidas en el segundo enlace si $C_B > C_A$ serán nulas. Si $C_A - C_B > \mu$ se tendrá un error en la aproximación por el sistema ficticio que aumenta al disminuir dicha diferencia. El sistema ficticio puede llegar a predecir un valor de pérdidas y las pérdidas reales ser casi nulas (como sería el caso por ejemplo cuando $C_B \rightarrow C_A$ y el tráfico es exponencial).

Resumiendo para el primer ejemplo visto: se verificó que siempre la región de aceptación del sistema ficticio está contenida en la del sistema real, que ambos sistemas son iguales si la capacidad de segundo enlace es menor o igual que la capacidad remanente del primer enlace (capacidad del primer enlace menos la media del flujo del tráfico restante), y que en este caso el error crece cuando esta diferencia se achica.

6.5.3. El análisis de una red más general

Veamos ahora un caso más interesante, como el que se muestra en la figura 6.13. En esta red se tienen dos niveles y consideraremos uno de los enlaces del segundo nivel.

En ella al enlace $M + 1$, llegan M agregados de flujo proviniendo cada uno de M enlaces

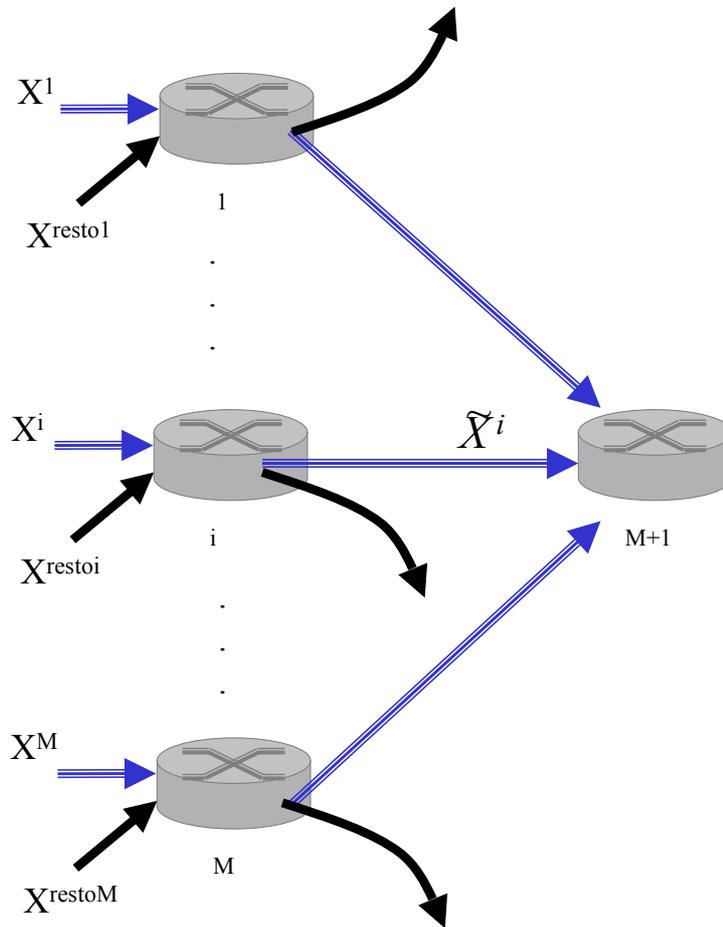


Figura 6.13: Análisis red general

anteriores. Llamamos $\tilde{X}^1, \dots, \tilde{X}^M$ a dichos agregados de flujo al arribar al enlace $M+1$ y X^1, \dots, X^M a los agregados al ingreso a los enlaces $1, 2, \dots, M$. Cada uno de dichos agregados de flujo en sus respectivos enlaces anteriores compartieron el enlace con un conjunto de otros agregados de flujo que identificamos $X^{\text{resto}i}, i = 1, \dots, M$. Es de hacer notar que este análisis permitirá sacar conclusiones sobre una red general con dos niveles, pero veremos al final que a partir de este estudio se pueden generalizar los resultados a una red con más niveles.

Analicemos el enlace $M+1$. En el sistema ficticio la función velocidad del gran desvío

de la probabilidad de pérdida viene dada por:

$$\mathbf{I}_{M+1}^F = -\inf\left\{\sum_{i=1}^M I_1^{X^i}(x_i) : x = (x_1, \dots, x_M), \sum_{i=1}^M x_i = C_{M+1}\right\} \quad (6.5.11)$$

La igualdad se cumple ya que la función \mathbf{I}_{M+1}^F es convexa, por ser suma de funciones convexas. En cambio en el sistema real la función velocidad viene dada por

$$\mathbf{I}_{M+1}^R = -\inf\left\{\sum_{i=1}^M (I_1^{X^i}(x_i) + I_1^{X_{resto}^i}(x_{resto_i})) \quad (6.5.12)$$

$$: x = (x_1, \dots, x_M, x_{resto_1}, \dots, x_{resto_M}), \sum_{i=1}^M \frac{x_i C_i}{\max(C_i, x_i + x_{resto_i})} > C_{M+1}\right\} \quad (6.5.13)$$

Lo que interesa es ver en qué condiciones el error cometido al analizar el sistema ficticio en lugar del real es nulo y cuando no. Sea μ_i el valor medio del tráfico X^i y μ_{resto_i} el del tráfico X_{resto}^i .

Para que las soluciones a ambos sistemas coincidan se debe cumplir que:

$$x_1 + x_2 + \dots + x_M = C_{M+1} \quad (6.5.14)$$

$$I_1^{X_{resto}^i}(x_{resto_i}) = 0 \quad \forall i \quad (6.5.15)$$

$$\frac{\partial I_1^{X^i}}{\partial x_i} = \frac{\partial I_1^{X^j}}{\partial x_j} \quad \forall i, j \quad (6.5.16)$$

La última condición se deduce de aplicar el método de los multiplicadores de Lagrange a la optimización del sistema ficticio. La condición 6.5.15 implica que el punto de funcionamiento del sistema real tendrá que ser $x = (x_1, \dots, x_M, \mu_{resto_1}, \dots, \mu_{resto_M})$ ya que $I_1^{X_{resto}^i}(\mu_{resto_i}) = 0$ para todo i .

Además para que se cumpla la condición 6.5.14, se debe verificar que

$$x_i + x_{resto_i} \leq C_i, \quad \forall i \in (1, \dots, M) \quad (6.5.17)$$

porque si no se cumple esto para algún i , entonces el punto de operación estará en $x_1 + x_2 + \dots + \frac{x_i C_i}{x_i + x_{resto_i}} + \dots + x_M = C_{M+1}$ y este punto tiene una función velocidad mayor que la del sistema ficticio.

De los puntos anteriores llegamos a la condición siguiente para el punto de operación:

$$x_i \leq C_i - \mu_{resto_i}, \forall i \in (1, \dots, M) \quad (6.5.18)$$

por lo tanto la condición para que ambos sistemas coincidan es que el punto $x^* = (x_1^*, \dots, x_M^*)$ que es solución de 6.5.16 verifique que

$$x_i^* \leq C_i - \mu_{resto_i}, \forall i \in (1, \dots, M) \quad (6.5.19)$$

Esta es una condición interesante pero se busca alguna condición que brinde un criterio más práctico. Para eso se puede observar que es necesario para que cumpla que las derivadas parciales sean todas iguales dos a dos que al menos todas tengan que el mismo signo y por lo tanto se cumple para cada una de ellas que el punto $x^* = (x_1^*, \dots, x_M^*)$ que es solución de 6.5.16 deberá cumplir $\forall i$ que

$$\mu_i \leq x_i^* \leq C_{M+1} - \sum_{j=1, j \neq i}^M \mu_j \quad (6.5.20)$$

Esto brinda una condición suficiente para que ambos sistemas se comporten igual (pueda eliminar los nodos anteriores) y es que:

$$C_{M+1} - \sum_{j=1, j \neq i}^M \mu_j \leq C_i - \mu_{resto_i}, \forall i \in (1, \dots, M) \quad (6.5.21)$$

Esto dice que para un LSP dado, si la capacidad equivalente libre en cada enlace para ese LSP (Capacidad del enlace menos la media de los flujos que comparten con él cada enlace) es constante o decreciente entonces puede usarse la red ficticia para calcular las pérdidas en cada enlace si cometer error. Sin embargo de la ecuación 6.5.20 sabemos que esta condición es suficiente pero no necesaria. Por lo tanto la capacidad puede ser 'algo' mayor e igual

cumplirse que ambos sistemas coincidan. ¿Cuánto mayor puede ser la capacidad remanente en un enlace y ambos sistemas coincidir? depende del punto donde todas las derivadas parciales de las funciones de velocidad I^{X_i} sean iguales y esto depende de la relación entre las diferentes funciones de velocidad que comparten los enlaces con el flujo en cuestión.

6.6. Conclusiones

En este capítulo se vio un modelo (muchas fuentes y buffer pequeño) que es adecuado para analizar una red MPLS. En este modelo se basará la propuesta de arquitectura para realizar ingeniería de tráfico en MPLS. Se ha visto una expresión que permite para calcular las pérdidas en cada enlace de la red y el ratio de pérdidas de punta a punta de un LSP. Esta es una métrica cóncava lo que permitirá utilizarla para modificar el algoritmo CSPF como ya se mencionó. Además Mazumdar demuestra que la región de aceptación para la red ficticia mencionada está contenida en la de la red real. Esto permitirá estudiar la red ficticia (más simple), y obtener en ella condiciones de aceptación que aplicadas a la red real aseguran que los requerimientos de QoS de cada clase de tráfico se cumplen.

Posteriormente estudiamos en qué condiciones la red real y la ficticia dan los mismos resultados respecto de las pérdidas en cada enlace de la red. Hemos encontrado una condición que de verificarse asegura que el análisis de pérdidas en ambas redes coincide. Su importancia radica en saber cuando el análisis realizado en la red ficticia es exacto, cuando no, en qué enlaces se comete error y de qué depende la magnitud de estos. Hay que hacer notar que aún cuando no coincidan en todos los enlaces las pérdidas igual las regiones de aceptación pueden coincidir. Cabe observar que como el ratio de pérdidas es una métrica cóncava, el enlace que importa para la QoS de punta a punta es aquel donde la función velocidad es más restrictiva (el valor de la $I(x)$ es menor). Si en estos enlaces coinciden las funciones $I(x)$, la región de aceptación podrá ser la misma aún cuando no coincidan en todos los enlaces de la red. En los enlaces donde la condición que encontramos no se cumpla (aquellos en que

hay mucha más capacidad remanente para el LSP que en los anteriores) se sobreestimarán las pérdidas. Cómo en ese enlace la capacidad remanente es grande para el agregado de flujos del LSP, el valor de $I(x)$ en dicho enlace será probablemente más grande que en los que dispone de una capacidad remanente menor y no será el enlace que importe para definir el ratio de pérdidas del LSP. Es decir que aún cuando en estos enlaces se sobreestimen las pérdidas, esta sobreestimación probablemente no tenga efecto respecto del ratio de pérdidas del LSP y la región de aceptación, obviamente si afectará si lo que se quiere es calcular las pérdidas de ese enlace.

La condición encontrada permite reconocer en cuales enlaces se está cometiendo error al usar el sistema ficticio y cual es la magnitud de ese error. El error será mayor cuanto más grande sea la capacidad remanente en el enlace $n + 1$ respecto del n en el camino del LSP.

En todos los casos aún cuando las condiciones encontradas no se cumplan sabemos que lo que puede suceder es que se sobreestime el ratio de pérdidas, pero nunca se subestimaré. Esto será una de las bases de la arquitectura propuesta.

Capítulo 7

Estimación del ancho de Banda Efectivo, el punto de operación y los parámetros de Calidad de Servicio de un enlace

7.1. Introducción

En los capítulos anteriores hemos visto como calcular la probabilidad de pérdida de un enlace y el ratio de pérdidas de punta a punta de una red. Se han visto las ecuaciones teóricas que permiten este cálculo en el régimen asintótico de muchas fuentes y en el de muchas fuentes y buffer pequeño. En ambos casos para el cálculo de los parámetros de calidad de servicio 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 5.2.2.

En el régimen asintótico de muchas fuentes (figura 7.1) la probabilidad de pérdida viene dada por la optimización de la fórmula *inf sup* (5.2.23). 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, pero si el supremo de dicha ecuación.

Para resolver dicha ecuación se debe realizar una doble optimización (en el parámetro

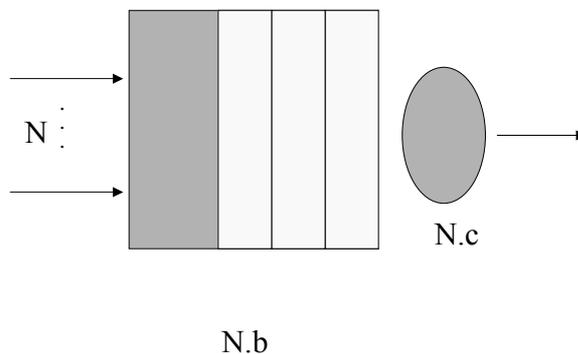


Figura 7.1: Regimen de muchas fuentes

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 casos reales en general no se cuenta con una fórmula teórica exacta para el ancho de banda efectivo.

En el caso general, se cuenta con trazas de tráfico y se debe resolver la ecuación (5.2.23), 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.

7.2. 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 de 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 calculando 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. Además es difícil encontrar un TCL para el ancho de banda efectivo que permita construir un intervalo de confianza del estimador. Veremos primero el enfoque no paramétrico y luego el paramétrico.

7.2.1. Estimadores no paramétricos

En este caso el estimador más comúnmente utilizado es el que estima el ancho de banda efectivo usando 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 [11], [36]:

$$\alpha_n(s, t) = \frac{1}{st} \log \left[\frac{1}{\lfloor T/t \rfloor} \sum_{j=1}^{\lfloor T/t \rfloor} e^{s\tilde{X}_j} \right] \quad (7.2.1)$$

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.

7.2.2. Estimadores paramétricos

El estimador paramétrico más simple asume que las fuentes son Poisson y luego estima el parámetro λ 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 de la fuente de tráfico 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.

Veremos a continuación el estimador para el caso continuo y más adelante veremos en un ejemplo el estimador en tiempo discreto.

Estimador markoviano de tiempo continuo

En este caso 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 [26],[28] :

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

donde $\bar{\mathbf{1}}$ 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 [35], 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.

7.2.3. 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. ($h_1 = 0$).

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 (7.2.2) se muestra en la figura 7.2.

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

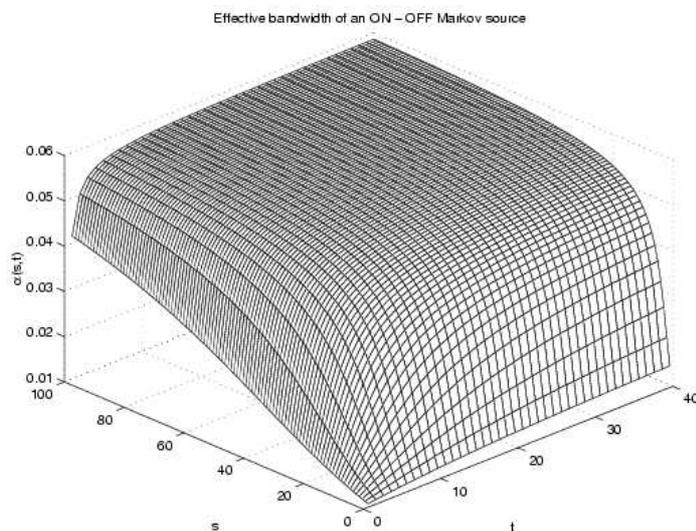


Figura 7.2: Ancho de banda efectivo para una fuente fluida markoviana

la figura 7.3 se muestra el ancho de banda efectivo para $t = 1$ fijo y en la figura 7.4 para $t = 50$ fijo. Como se puede apreciar en la primer figura 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 necesitamos una traza muy larga para tener una buena aproximación. El problema se detecta en valores de s grande es decir para valores cercanos al pico de la fuente. Al no ser la traza lo suficientemente larga, la sucesión de muchos picos no ocurre y en esos valores se estima mal.

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

En esta sección veremos la aplicación de los estimadores paramétricos markovianos a una traza de tráfico real. En esta parte de la tesis se busca la posibilidad de utilizar el estimador paramétrico markoviano para trazas de tráfico reales. Como se verá más adelante

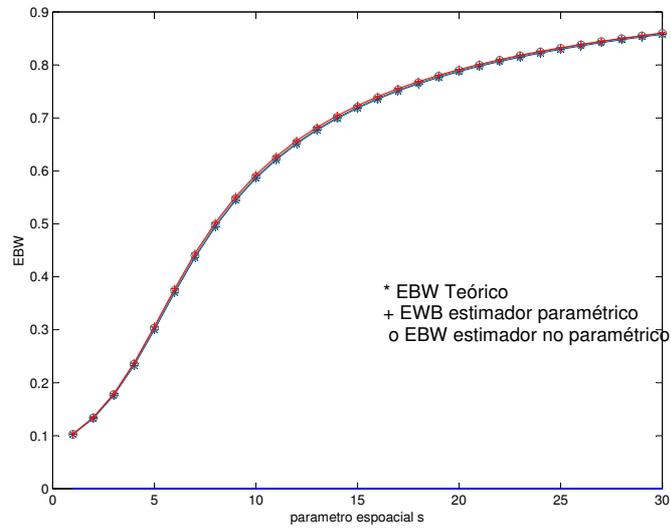


Figura 7.3: Ancho de banda efectivo teórico y estimado para una fuente fluida markoviana para $t=1$ fijo

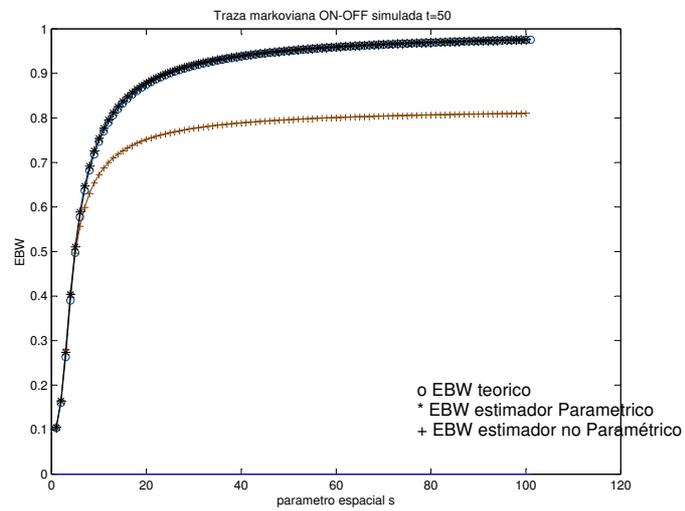


Figura 7.4: Ancho de banda efectivo teórico y estimado para una fuente fluida markoviana para $t=50$ fijo

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 7.5, 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, usaremos 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, vimos 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 \mathbb{P} , la matriz de transición de estados. Más precisamente se estimará la ley límite invariante que llamaremos π_∞ . Se asume que la cadena es homogénea, regular y de espacio de estados finito. Por lo tanto existe una ley límite invariante. Se puede probar [15], el siguiente Teorema:

Teorema 7.1. *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 definimos*

$$\mu_\infty(f) = \sum_{x \in E} f(x) \pi_\infty(x) \quad (7.2.2)$$

el valor esperado de f respecto de la probabilidad π_∞ . se cumple que

$$\mathbf{E}(f(X)) \xrightarrow[n]{c.s.} \frac{1}{n} \sum_{i=1}^n f(x_i) \xrightarrow[n]{c.s.} \mu_\infty(f) \quad (7.2.3)$$

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

$f(x) = \exp(sh_x)$ se obtiene que

$$\alpha(s, 1) = \frac{1}{s} \log \sum_{x \in E} (\pi_\infty(x) \exp(sh_x)) \quad (7.2.4)$$

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 n el largo total de la traza.

Al tratar de aplicar este estimador en la práctica sobre la traza mpeg surgieron una serie de problemas. En particular para calcular n_x debemos decidir sobre la traza si cada punto pertenece a un estado o a otro. Para poder tomar estas decisiones 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 dijimos: 'conocidos los h_x ', pero *¿cuál es el nivel h_x de emisión en cada uno de los estados?*. En el caso ideal es nivel es constante pero si se aprecia la traza mpeg en estudio se puede ver que en el caso real no 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.

Veamos 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 donde 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 un estado u a

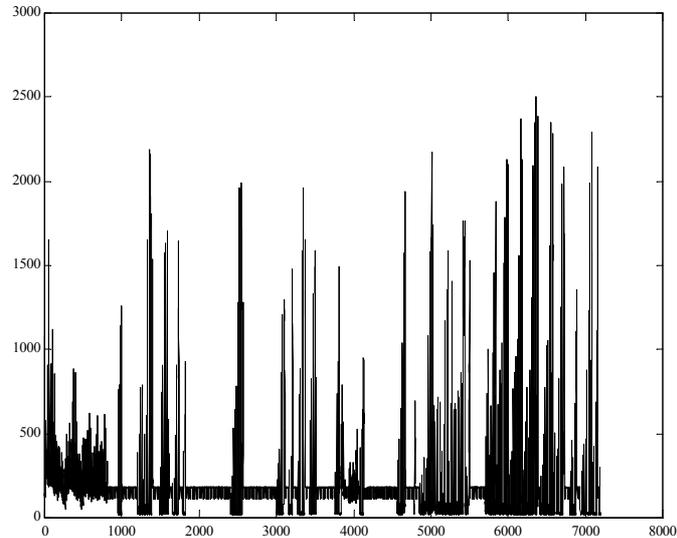


Figura 7.5: Traza de tráfico mpeg

otro. En la figura 7.6 se muestra que dependiendo de la elección de umbral el ancho de banda efectivo estimado puede variar mucho.

Para responder a las preguntas anteriores, en la tesis proponemos un algoritmo basado en el siguiente test de markovianidad y en otras consideraciones que se realizarán más adelante. Asumimos que el espacio de estados E tiene e elementos y que la matriz de transición de estados \mathbb{P} es tal que tiene r elementos $p(i, j) > 0$ con $(i, j) \in E^2$. Se prueba [15] que

$$Z_n = \sum_{(i,j,k) \in E^3} \frac{(N_{ijk} - N_{ij}N_{jk}/n)^2}{N_{ij}N_{jk}/N_j} \xrightarrow{\text{ley}} \chi^2(re - r - r + e) \quad (7.2.5)$$

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 distribución de la variable aleatoria Z_n definida de esa manera para una cadena de Markov converge en ley cuando $n \rightarrow \infty$ a una distribución

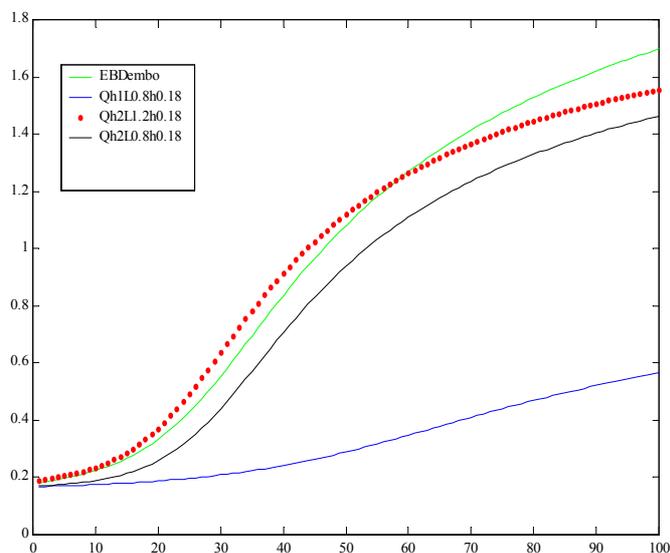


Figura 7.6: EBW con estimador markoviano variando el umbral, t fijo =1

chi cuadrado.

Si asumimos que todos los términos de la matriz de transición son positivos con el resultado anterior podemos testear la markovianidad de una traza con nivel asintótico α , 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

- Para número de estados = 2 hasta N
- Recorrer todos los umbrales posibles con un cierto paso dado y Calcular el Test de Markovianidad para cada caso
- Obtener el conjunto de umbrales para los cuales el test sea mínimo.
- Calcular probabilidad de acierto del modelo para el conjunto de umbrales que con este número de estados obtuvo el mejor test.
- estados = estados +1

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 mencionada se le aplicó el algoritmo anterior y se obtuvo:

- para dos estados: Umbral óptimo : 1.79 Mb/s, Chi2 = 0.35 (dos grados de libertad), $P(\text{acierto}) = 0.85$
- para tres estados: Umbrales óptimos: 1.5 y 1.84 Mb/s, Chi2 = 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, Chi2 = 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 proponemos 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 figuras 7.7 se muestra el pico del histograma en el rango inferior, por lo que se adoptó $h_0 = 200kb/s$ y en la figura 7.8 el histograma para los dos rangos superiores, por lo que se adoptó $h_1 = 1700kb/s$ y $h_2 = 2100kb/s$.

En la figura 7.9 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.

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

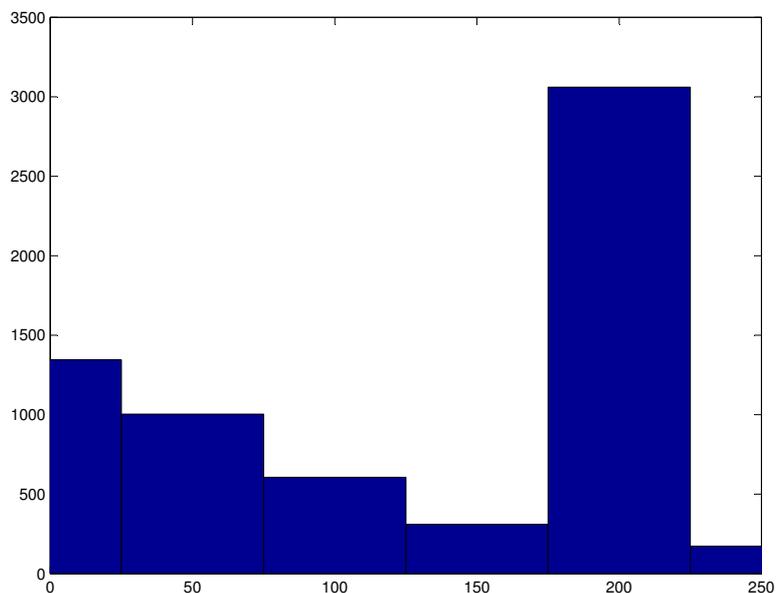


Figura 7.7: Histograma de la traza entre valores de (0 y 250 kb/s)

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. Por estas razones puede convenir ajustar un estimador paramétrico aún cuando tengamos uno no paramétrico. Con la justificación anterior se podría realizar un ajuste de las velocidades h_i entorno a las que se obtienen del histograma buscando que el EBW se pareciera lo más posible al del estimador paramétrico, con algún criterio como minimizar (globalmente o en algún rango de s y t) el máximo de la diferencia o el área entre las curvas.

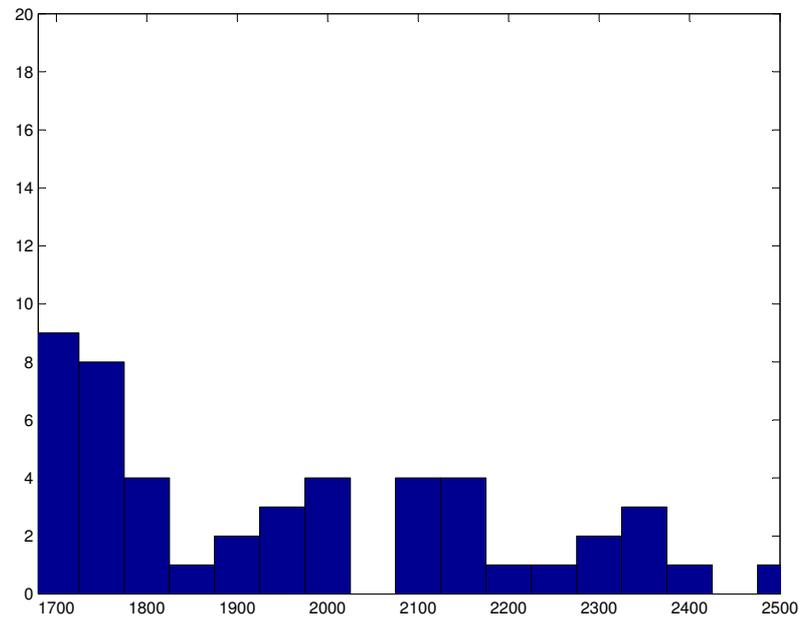


Figura 7.8: Histograma de la traza entre valores de 1700 y 2500 kb/s

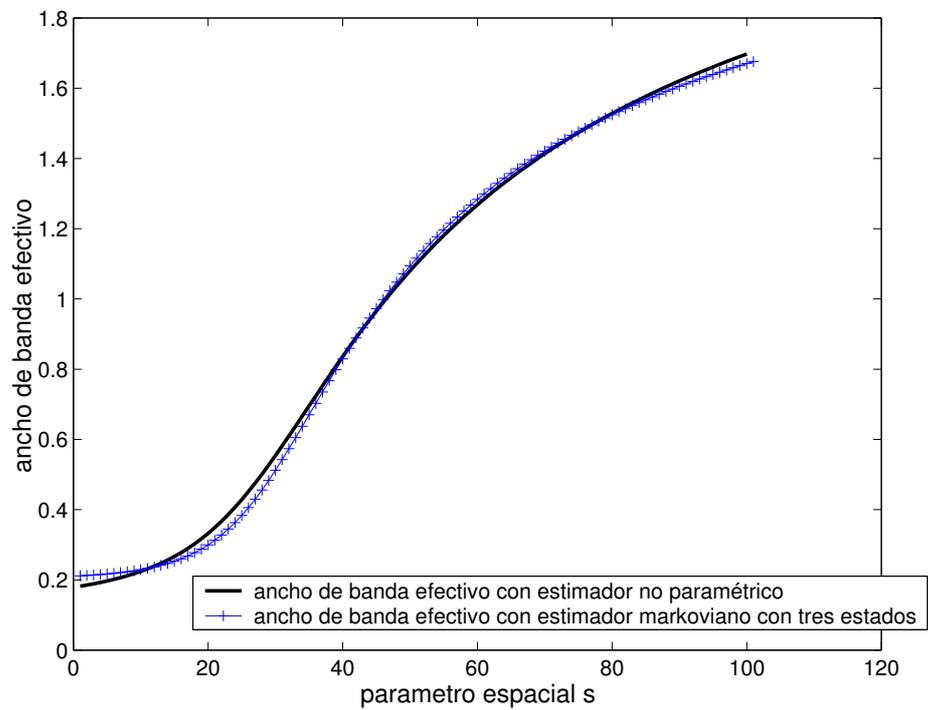


Figura 7.9: Comparación del EBW estimado de la traza

7.3. 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 cuando se estima el punto de operación (s_n^*, t_n^*) de la ecuación *inf sup* se obtendrá un estimador consistente de (s^*, t^*) y si será valido algún TCL para el punto de operación. La respuesta de esta pregunta se desarrolló en el un trabajo que realizamos en conjunto con otros integrantes del grupo ARTES [2]. El principal resultado del citado trabajo es el siguiente teorema:

Teorema 7.2. *Sea $\alpha(s, t) = \log \Lambda(s, t)$. Si $\Lambda_n(s, t)$ es un estimador de $\Lambda(s, t)$ tal que ambos son funciones C^1 y:*

$$\Lambda_n(s, t) \xrightarrow[n]{} \Lambda(s, t) \quad (7.3.1a)$$

$$\frac{\partial}{\partial s} \Lambda_n(s, t) \xrightarrow[n]{} \frac{\partial}{\partial s} \Lambda(s, t) \quad (7.3.1b)$$

$$\frac{\partial}{\partial t} \Lambda_n(s, t) \xrightarrow[n]{} \frac{\partial}{\partial t} \Lambda(s, t) \quad (7.3.1c)$$

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

$$b + ct_n^* - \frac{\frac{\partial}{\partial s} \Lambda_n(s_n^*, t_n^*)}{\Lambda_n(s_n^*, t_n^*)} = 0 \quad (7.3.2a)$$

$$cs_n^* - \frac{\frac{\partial}{\partial t} \Lambda_n(s_n^*, t_n^*)}{\Lambda_n(s_n^*, t_n^*)} = 0 \quad (7.3.2b)$$

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:

$$\sqrt{n} ((s_n^*, t_n^*) - (s, t)) \xrightarrow[n]{w} N(\vec{0}, \Sigma) \quad (7.3.3)$$

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

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.

Entendemos por 'buen' estimador aquel que verifica las hipótesis del teorema anterior. Como se observa en dicho trabajo los estimadores mencionados en este capítulo del ancho de banda efectivo (el no paramétrico y los markovianos) verifican estas hipótesis.

Para verificar el resultado del teorema anterior se realizó el análisis 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 (5.2.24). En el caso que se está analizando se resolverá dicha optimización con $\alpha(s, t)$ de la ecuación teórica (7.2.2) 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.

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

Para resolver la ecuación (5.2.24) 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 simularon trescientas trazas $K = 300$, de largo $T = 100000$ muestras y se construyó, para cada traza simulada indexada

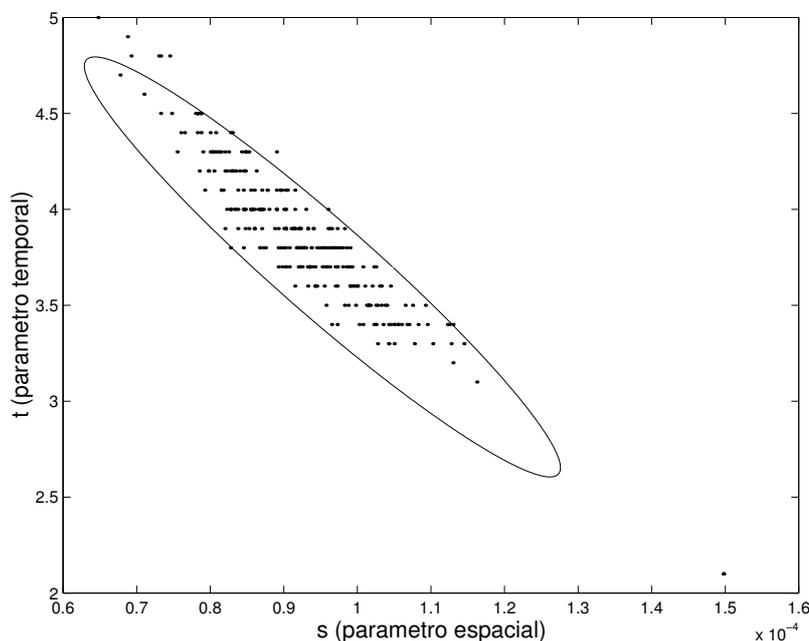


Figura 7.10: Punto de operación estimado y región de confianza con nivel 95 %

por $i = 1, \dots, K$ el correspondiente estimador $(s_n^*(i), t_n^*(i))$. Por el teorema 7.2 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 Σ . Se estimó la matriz Σ usando las covarianzas empíricas

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

dad 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, podemos decir que aproximadamente:

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

de donde un nivel α 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 Σ_K . Se construyó entonces la región de confianza de nivel 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 7.10, donde se verificó que el 95,33 % de los puntos estimados se ubicaron dentro de la región predicha.

7.4. Estimación de los parámetros de calidad de servicio de un enlace

Como mencionamos, la estimación del punto de operación tiene como objetivo estimar la probabilidad de pérdida del enlace según la ecuación (5.2.24) 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 [40]. 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 7.2 entonces el estimador

$$\gamma_n = \inf_t \sup_s ((ct + b)s - st\alpha_n(s, t)) \quad (7.4.1)$$

es consistente y verifica el TCL .

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

$$q_n = P_n(Q_N > B) \approx \exp^{-N\gamma_n} \quad (7.4.2)$$

siendo Q_N el tamaño de la cola y N la cantidad de fuentes del sistema. En la figura 7.11 se muestra la estimación de γ_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%.

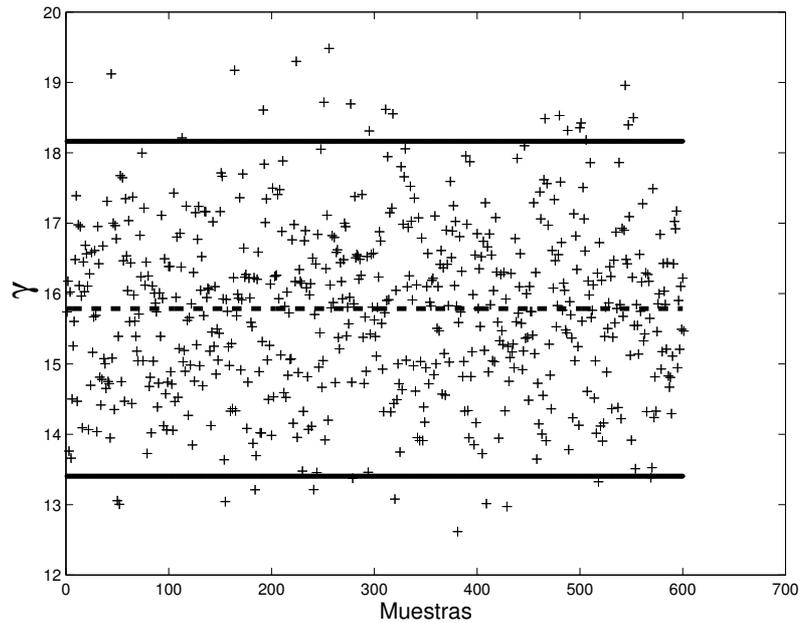


Figura 7.11: γ_n estimado, γ teórico e intervalo de confianza de 95%.

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

Por último los resultados anteriores pueden extenderse al caso del diseño de un enlace de una red, al que se le requiere cierto nivel de calidad de servicio. Es decir, 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). Se procede análogamente si se tiene la misma información que antes pero se define 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 *inf sup* vista antes.

El tamaño de buffer mínimo para asegurar probabilidad de pérdida γ está dada por la ecuación [11]:

$$B_n = \sup_t \inf_s (G_n(s, t)) \quad (7.4.3)$$

$$G_n(s, t) = \frac{(Nst\alpha_n(s, t) + N\gamma)}{s} - Ct \quad (7.4.4)$$

y la capacidad mínima necesaria para asegurar probabilidad de pérdida γ es:

$$C_n = \sup_t \inf_s (K_n(s, t)) \quad (7.4.5)$$

$$K_n(s, t) = \frac{(Nst\alpha_n(s, t) + N\gamma)}{st} - B/t \quad (7.4.6)$$

En las figuras 7.12 y 7.13 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.

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.

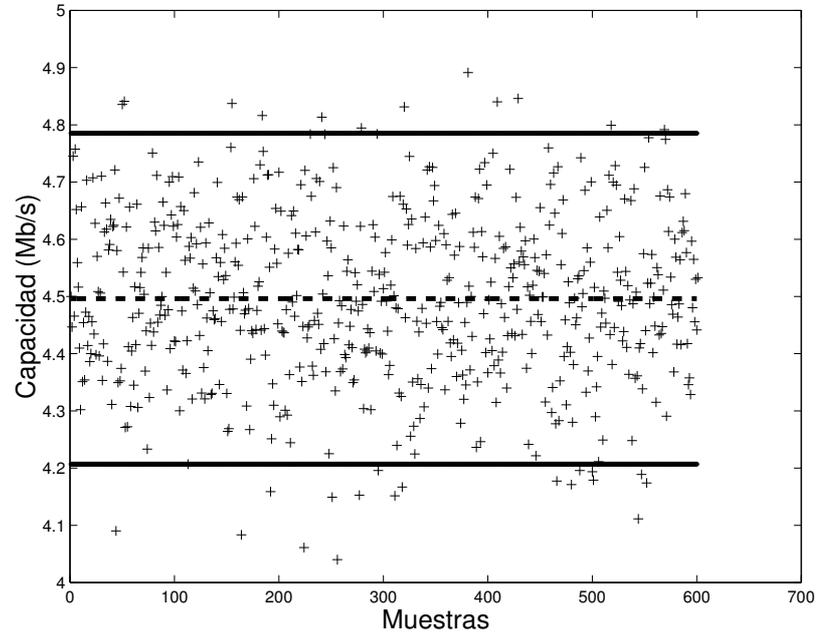


Figura 7.12: Capacidad estimada, capacidad teórica e intervalo de confianza de 95 %.

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 7.14 se 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 α (por ejemplo α teórico y un α_n estimado). Si se pasa de una a otra manteniendo B fijo, γ varía muy poco. En cambio, si se mantiene γ fijo, B varía mucho.

De manera similar al caso anterior se puede analizar la variación de γ con la capacidad

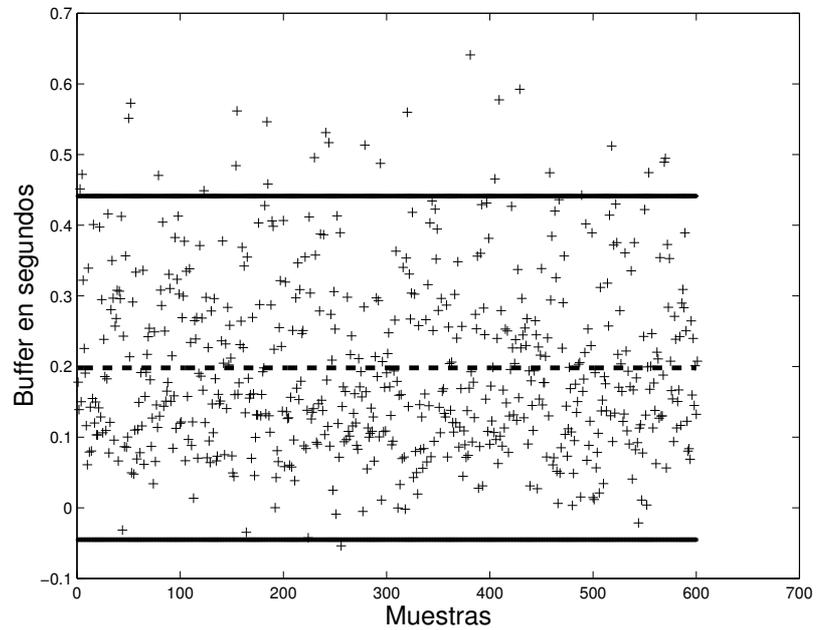


Figura 7.13: Tamaño de buffer estimado, teórico e intervalo de confianza de 95 %.

del enlace C . En la figura 7.15 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 γ varía poco. A diferencia del caso anterior, si se mantiene γ fijo C también varía poco. Esto explica la poca dispersión de C en la figura 7.12.

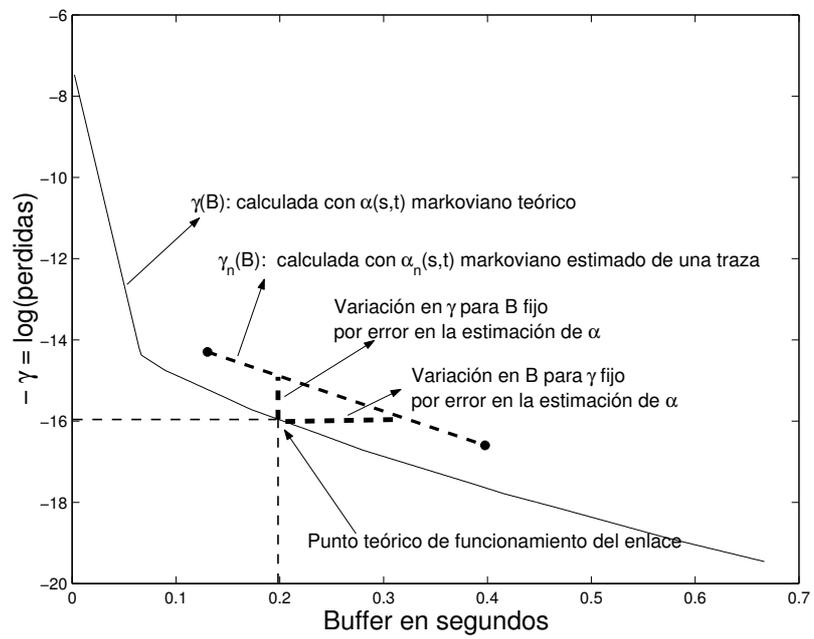


Figura 7.14: Variación de $-\gamma$ con el tamaño de buffer

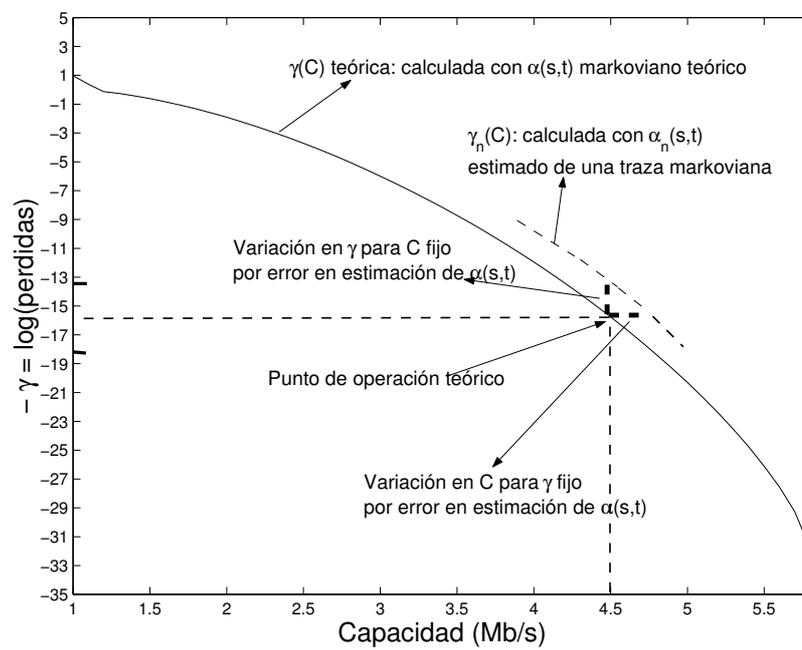


Figura 7.15: Variación de $-\gamma$ con la capacidad del enlace

Capítulo 8

Una propuesta para realizar Ingeniería de Tráfico y garantizar Calidad de Servicio en MPLS

8.1. Introducción

En este capítulo se proponen algunas herramientas que permiten en una red MPLS asegurar la QoS requerida a cada agregado de flujos, haciendo un uso eficiente de los recursos de la red. La idea es usar los conceptos y resultados vistos en los capítulos anteriores para realizar Ingeniería de Tráfico en una red MPLS. En estos últimos años se han propuesto diversos mecanismos para realizar ingeniería de tráfico en MPLS. Existen dos aspectos que se abordarán en esta propuesta y que son importantes a la hora de realizar ingeniería de tráfico en MPLS:

- ¿Cómo establecer los caminos en la red (LSPs)? Este problema ha sido abordado a través del ruteo basado en restricciones (CBR). La idea de CBR es establecer caminos asegurando que verifiquen las condiciones de QoS exigidas para los diferentes tipos de tráfico que atraviesa la red MPLS.
- Una vez establecidos los LSPs, cómo repartir la carga entre los diferentes caminos establecidos.

Muchas de las propuestas realizadas para los puntos antes mencionados se basan en optimizaciones realizadas fuera de línea. En éstas se obtiene una configuración de la red que se establece y se pone a operar. Para una red con un tamaño importante, no es posible re-optimizar en línea cada vez que el volumen de tráfico que atraviesa la red cambia. La idea de la propuesta es hacer Ingeniería de Tráfico en línea para adaptar la red a los cambios que surjan, de forma que se sigan cumpliendo las restricciones de QoS y aunque el resultado que se obtenga no sea el óptimo. Se entiende que una propuesta de ingeniería de tráfico para ser realmente eficiente debe ser de tipo híbrido: Algoritmos que establezcan fuera de línea una configuración inicial y algoritmos que permitan en línea ajustarse a la dinámica del tráfico y de la red sin minimizando las re-configuraciones fuera de línea.

Para trabajar en línea se deben buscar algoritmos simples, que permitan tomar decisiones de manera eficaz. Si los cambios son de tal magnitud que estos algoritmos no pueden encontrar una solución, se deberá notificar al sistema de gestión de la red y probablemente sea el momento de correr una nueva optimización de la red fuera de línea. En la propuesta se ha buscado una solución que cumpla al menos con los siguientes criterios que son importantes a la hora de realizar ingeniería de tráfico en línea:

- Los algoritmos a utilizar deben ser simples. Parte de la propuesta se basa en realizar algunas modificaciones a los algoritmos de ingeniería de tráfico en línea disponibles en routers comerciales. En particular proponemos modificaciones al algoritmo CSPF (Constrained Shortest Path First) disponible en routers CISCO y JUNIPER por ejemplo ([27],[21]). La idea es modificar este algoritmo e integrarlo con algoritmos simples de balanceo de carga. Se procurará además, no aumentar demasiado la complejidad de los algoritmos utilizados hoy. Este último punto es importante para que puedan ser implementados sin cambios mayores. El objetivo de un ISP no es tener una red con un algoritmo óptimo sino tener una red simple en gestión y operación, que utilice lo mejor posible los recursos y que satisfaga restricciones en los parámetros de QoS de

los servicios ofrecidos.

- Los parámetros del tráfico a medir en línea deben ser los menos posibles, y con algoritmos sencillos. Muchas propuestas existentes se basan en medir en línea retardos o pérdidas en todos los nodos de la red, lo cual presenta diversas dificultades de implementación, que hacen difícil su aplicación [31].
- Se debe seguir el criterio que ha sido uno de los pilares del crecimiento de Internet: Si hay que agregar funcionalidades a los routers, se debe procurar que estas se concentren en la frontera de la red, manteniendo los enrutadores internos lo más simples posible. Una de las dificultades que existe actualmente para desplegar la arquitectura DiffServ por ejemplo, es que en los nodos interiores se deben configurar políticas sofisticadas de scheduling y dropping. Esto se torna complejo en la práctica y conlleva modificaciones importantes de configuración si se producen cambios en la red. Buscaremos una arquitectura donde la configuración de los nodos interiores sea lo más simple posible para que la red sea lo más escalable posible.
- Relacionado con el punto anterior encontramos el próximo criterio: Si hay que hacer cambios en la configuración de los enrutadores, se debe buscar una solución que modifique la menor cantidad de enrutadores posible y los cambios en éstos debieran ser los mínimos posibles.
- El sistema debe ser estable. Es decir se deben procurar algoritmos que no generen oscilaciones del tráfico de la red, como sucede con muchos algoritmos de ruteo basados en calidad de servicio.
- Utilizar primero los cambios menos costosos para la operación y gestión de la red. Por ejemplo, si se puede ajustar una configuración re-enrutando tráfico por LSPs existentes, esto es preferible a establecer un nuevo LSP para canalizar dicho tráfico.

- Tener en cuenta consideraciones administrativas como por ejemplo que no se desea que cierta clase de tráfico utilice ciertas zonas de la red.
- Tener en cuenta algún modelo de negocios que beneficie al ISP al aplicar estas políticas. El objetivo de un ISP, para desplegar herramientas de QoS e ingeniería de tráfico es aumentar los beneficios de su negocio. El esquema de negocios que se encuentra hoy en la mayoría de los ISPs, se basa en el cobro de una tarifa uniforme sin distinguir servicios y sin ofrecer ningún tipo de garantías respecto del servicio de transporte ofrecido.

Para que estas propuestas u otras tengan sentido deben tener en cuenta algún modelo de tarificación sobre el servicio de transporte diferente al actual.

8.2. La arquitectura de red MPLS propuesta para ofrecer QoS

En la propuesta de la tesis se asume que el backbone MPLS cumple las hipótesis de 'muchas fuentes' y buffer pequeño. En este modelo se utilizará como parámetro de QoS a controlar el ratio de pérdidas visto en el capítulo 6. Como es un backbone trabajando con el modelo de 'muchas fuentes' y 'buffer pequeño' si se controlan los paquetes perdidos, el retardo de cola estará acotado por un valor máximo que será 'pequeño' en cada enlace. El control del retardo de punta a punta se traduce entonces en controlar que la cantidad de enlaces que atraviesa un LSP no supere una cierta cota que dependerá de la clase.

En esta tesis se denominará 'clase' a cada tipo de servicios que el ISP ofrece y sobre los que asegura ciertos niveles de calidad. Es decir que el ISP asegurará por ejemplo ciertos niveles de calidad de servicio (en pérdidas y/o retardo) al tráfico de VoIP que quiera usar su red, ciertos niveles para el tráfico de vídeo conferencia, etc. El primer punto a definir es el criterio con el que se asignan las FECs en cada nodo de entrada a la red. Se propone que el agregado de flujos en FECs se realice por las clases de servicios y por el par origen-destino

en la nube MPLS. Por ejemplo que en cada LER de la frontera se agregará en una FEC el tráfico de VoIP que ingresa por ese LER y sale por otro LER de la red, en otra FEC para el mismo par de LERs el tráfico de video conferencia, etc. Esto permitirá asumir que las diferentes fuentes de tráfico de una FEC tienen el mismo modelo estadístico.

Un comentario sobre este punto es que si bien se asume que las clases son uniformes esto no necesariamente quiere decir que todas las clases pertenecen a un único tipo de fuente. Por ejemplo en un modelo se podrán considerar las siguientes clases: 1. VoIP 2. Video 3. Peer to Peer 4. http, mail, ftp Este último tiene tráfico de fuentes con diferentes estadísticas (http,ftp, mail). Se podrán agrupar en una única clase sí es posible asumir que: a) Las garantías de QoS sobre todos esos tipos de fuentes son las mismas. b) Que las proporciones de c /tipo de fuente dentro de la clase se mantienen constantes. Por ejemplo, asumamos que fuera de línea se observó que aproximadamente se tiene un 10 % de ftp, un 30 de mail y un 60 de http. Lo que se debe cumplir es que si en línea se produce un aumento de un 15 % del tráfico de la clase 4, se pueda asumir que las proporciones de cada tipo de fuente se siguen manteniendo.

8.3. Mediciones de tráfico a realizar en el modelo propuesto

Como se mencionó en la sección anterior el modelo propuesto implica que todos los flujos que circulan por un LSP tienen el mismo modelo estadístico. Esto quiere decir que el ancho de banda efectivo del agregado de flujos que atraviesa dicho LSP ($\alpha_{LSP}(s, t)$) lo podemos representar como:

$$\alpha_{LSP}(s, t) = N_{LSP}\alpha_C(s, t) \quad (8.3.1)$$

siendo $\alpha_C(s, t)$ el ancho de banda efectivo de una fuente tipo de esa clase y N_{LSP} la cantidad de fuentes del agregado de flujo que se enruta sobre ese LSP. Por lo tanto si los flujos que atraviesan ese LSP aumentan en un 20 %, y llamamos $\alpha_{LSP}^i(s, t)$ al ancho de banda efectivo

previo al aumento de tráfico, el ancho de banda efectivo del nuevo tráfico ($\alpha_{LSP}^f(s, t)$) será:

$$\alpha_{LSP}^f(s, t) = 1,2\alpha_{LSP}^i(s, t) \quad (8.3.2)$$

Por lo tanto las mediciones a realizar son las siguientes:

- Fuera de línea se miden trazas del tráfico que ingresa a cada LSP de la red y se estima el ancho de banda efectivo de este tráfico. Como se trabaja con buffer pequeño sólo interesará para la estimación la escala temporal $t = 1$. Este ancho de banda efectivo estimado fuera de línea corresponde a $\alpha_{LSP}^i(s, 1)$ en la ecuación 8.3.2
- Posteriormente en línea se mide el valor medio de dicho tráfico en la ventana de tiempo que corresponda. Como en cada LSP se asume que se agregan fuentes independientes e idénticamente distribuidas, con la estimación de la media se obtiene una estimación de la nueva función ancho de banda efectivo ($\alpha_{LSP}^f(s, t)$). Este ancho de banda efectivo corresponde al tráfico que circula por dicho LSP en ese momento. Con esta información se obtiene también una estimación del número de fuentes que están circulando por el LSP.

Es posible estimando el nuevo valor medio del tráfico por un LSP, estimar su función ancho de banda efectivo y la cantidad de fuentes que lo utilizan. Esta observación será usada más adelante para el análisis de los parámetros de QoS de punta a punta en la red. Es de hacer notar que la medición del valor medio del tráfico en los LSPs en el ingreso a la red es la única medición en línea que se realizará. Esto está de acuerdo con los criterios explicados antes.

8.4. Bases para calcular los parámetros de Calidad de Servicio de cada agregado en línea

El modelo de red MPLS explicado antes ('muchas fuentes' y 'buffer pequeño') se encuentra dentro de las hipótesis de la propuesta de Mazumdar et. al.[34], explicadas en el

capítulo 6. En ese capítulo se vio que la probabilidad de pérdida en cualquier nodo de la red tenía función buena de velocidad de grandes desvíos:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbb{P}(\text{perdidas en el enlace } k) = -\mathbf{I}_k = \quad (8.4.1)$$

$$- \inf \left\{ \sum_{m=1}^M I_1^{X^m}(x_m) : x = (x_m) \in \mathbb{R}^M, g(x) = \sum_{m=1}^M g_k^m(x) > C_k \right\} \quad (8.4.2)$$

Siendo $g_k^m(x)$, la función que relaciona la velocidad instantánea del tráfico de entrada del tipo m a un enlace interior k con la velocidad instantánea del tráfico de entrada a la red. Para el caso de una red feed-forward g_k^m se podía calcular como la composición iterativa de afuera hacia dentro de la red de las funciones:

$$f_k^m(x_1, \dots, x_M, C_k) = \frac{x_m C_k}{\text{máx}(\sum_{i=1}^M x_i, C_k)} \quad (8.4.3)$$

Por lo tanto, conociendo la función de velocidad de grandes desvíos de cada clase de tráfico a la entrada a la red (de cada LSP en MPLS) $I_1^{X^m}$, la capacidad de los enlaces y la topología lógica de la red, es posible conocer la probabilidad de pérdida en cualquier enlace. Además se vio que el ratio de pérdidas de punta a punta de la red cumplía la siguiente ecuación:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log \mathbf{L}^{m.N} = - \min_{k \in r^m} \mathbf{I}_k \quad (8.4.4)$$

y por lo tanto conociendo la información anterior (tráfico a la entrada, topología y capacidades) es posible conocer el ratio de pérdidas de punta a punta de cada LSP. Con esta información es posible entonces encontrar la región de aceptación de la red especificado el ratio de pérdidas admisible para cada tipo de flujo. Recordemos que la función de velocidad de grandes desvíos del tráfico de entrada es posible expresarla de la siguiente manera:

$$I_1^{X^m}(x) = \sup_{\lambda > 0} \lambda x - \lambda \alpha^{X^m}(\lambda, 1) \quad (8.4.5)$$

Siendo α^{X^m} , el ancho de banda efectivo del tráfico del LSP m en la entrada a la red. El procedimiento anterior, junto con las consideraciones realizadas en la sección , permite

midiendo la media en línea del tráfico de cada LSP a la entrada (según el procedimiento visto en dicha sección), estimar el ratio de pérdida de punta a punta de la red para cada LSP. De esta forma se podrían desarrollar algunas herramientas de Ingeniería de Tráfico. Sin embargo este procedimiento tiene aún dos aspectos que implican complejidades que dificultan su aplicación en línea:

- En una red compleja y donde además, aplicando herramientas de ingeniería de tráfico, se podría cambiar la configuración de LSPs o el reparto de tráfico que se envía por ellos, el cálculo iterativo en línea de las funciones $g_k(x)$ para cada enlace puede ser muy costoso.
- Es necesario también realizar en línea todas las optimizaciones de la ecuación 8.4.5 para cada enlace

Con el objetivo de simplificar estos dos aspectos para aplicar el procedimiento en línea se verán a continuación dos procedimientos que permiten algoritmos más simples, a cambio de sobre estimar el ratio de pérdidas y por lo tanto disminuir la región de aceptación. De todas formas, si bien el procedimiento no llevará al óptimo, en el sentido que se podrían hacer circular más conexiones que con el método propuesto, permitirá que entre re-optimizaciones o re-configuraciones mayores, hacer en línea modificaciones que permitan que más flujos puedan usar la red respetando sus requerimientos de calidad de servicio (cosa que siempre se respetará ya que los criterios que se seguirán estarán siempre sobreestimando las pérdidas y nunca subestimando). Por otro lado la metodología propuesta en caso que no se pudiera en algún aspecto asegurar calidad de servicio, será capaz de notificar a los operadores para que tomen acciones.

8.4.1. Cálculo de performance a través de la 'red ficticia'

Como vimos en la sección 6.5 se utilizará una aproximación para el estudio de un enlace interior de una red consistente en asumir que los enlaces anteriores no afectan la estadística

del tráfico que arriba a dicho enlace ('red ficticia'). Con esta hipótesis se puede calcular la performance de un enlace solamente conociendo el ancho de banda efectivo medido en el ingreso a la red de cada LSP que lo atraviesa. Se vio que esta aproximación es exacta en bajo ciertas condiciones. Cuando no es exacta también se observó, que siempre se está del lado seguro desde el punto de vista de la QoS.

8.4.2. Aproximación en línea del punto de operación

En la arquitectura planteada si se mide en línea el valor medio del tráfico en cada LSP, es posible calcular la probabilidad de pérdida en cada enlace y el ratio de pérdida del LSP al atravesar la red. Se presenta una dificultad: resolver en línea la optimización que permite calcular el punto de operación. Sobre todo que interesa no sólo calcular en un enlace los parámetros de QoS sino en toda la red. Esto lleva a que cuanto mayor sea la red más tiempo de cálculo es necesario y mayores las necesidades de almacenamiento de información. Para solucionar este problema, debemos simplificar los cálculos numéricos a realizar.

La propuesta es calcular fuera de línea el punto de operación de cada enlace($s_i^*, 1$) y luego utilizar dicho punto en línea para estimar las pérdidas. Esta aproximación siempre asegura que los requerimientos de QoS se cumplen. El costo que se paga es posiblemente sub-utilizar los recursos de la red, pero el objetivo no es obtener el punto de operación que optimice los recursos de la red, sino adaptar la red ante variaciones en línea. De esta forma se minimizarán las reconfiguración off-línea asegurando la QoS requerida y teniendo un buen aprovechamiento de recursos.

Para explicar por qué operando de esta manera siempre se puede asegurar que las pérdidas cumplen con los requerimientos de QoS, se verá a continuación cómo se realiza el cálculo del punto de operación.

Para eso se resuelve:

$$\sup_s (Cs - s\alpha_{LSP}^i(s, 1)) \quad (8.4.6)$$

y se obtiene el punto s_i^* . Si no se usa la aproximación mencionada se debería resolver nuevamente la ecuación con el nuevo valor medio medido. Sea M el incremento del valor medio, entonces se debe resolver

$$\sup_s (Cs - sM\alpha_{LSP}^i(s, 1)) \tag{8.4.7}$$

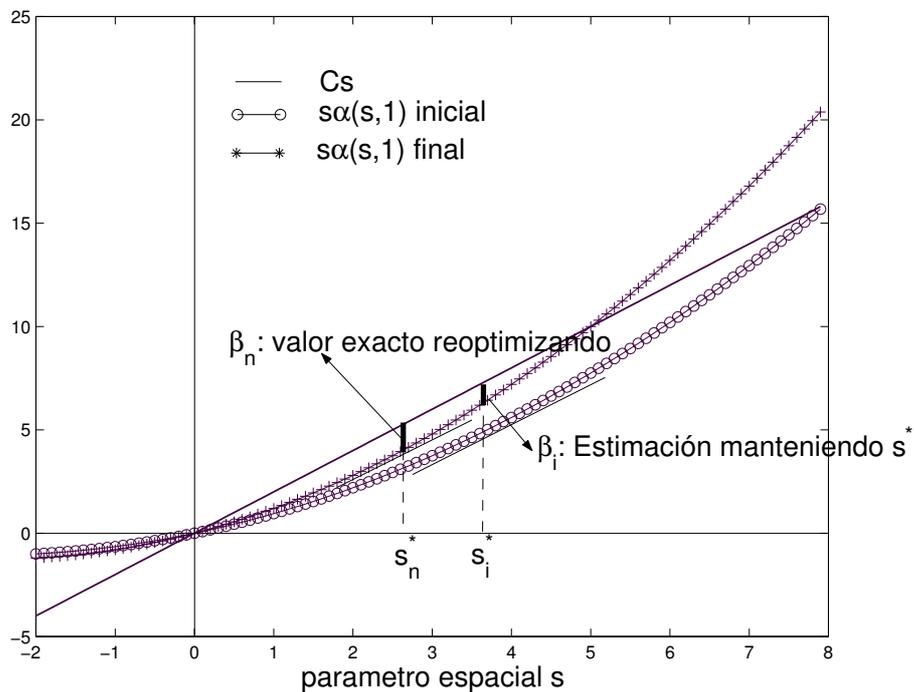


Figura 8.1: Solución de la ecuación de optimización en s

obteniendo un s_n^* . Gráficamente en la figura 8.1 se ve la resolución de esta ecuación al variar la cantidad de tráfico por el enlace. Como se ve el punto donde se da el supremo sufre una variación. Llamemos

$$\beta(s, 1) = (Cs - sM\alpha_{LSP}^i(s, 1)) \tag{8.4.8}$$

Si utilizamos el valor s_i^* original en lugar del real s_n^* , por ser este último el punto donde

se da el supremo de la ecuación anterior, se cumple que

$$\beta(s_n^*) \geq \beta(s_i^*) \quad (8.4.9)$$

Antes de extender estos resultados a la evaluación en toda la red, hay que comentar que en el caso que en un enlace confluyan múltiples LSPs no presenta mayores dificultades a las ya analizadas pues la fórmula se extiende como ya se mencionó a una expresión donde intervienen todas las clases de tráfico que atraviesan el enlace. Para analizar si se cumplen los requerimientos de QoS comprometidos para cada clase de tráfico en ese enlace, se verificarán las condiciones con los nuevos valores medios medidos del tráfico en cada LSP que atraviesa el enlace, y con el punto de operación calculado fuera de línea. Si cada LSP representa un porcentaje del tráfico total del enlace, entonces el impacto por la variación del tráfico en un LSP se verá atenuado, y la variación del punto de operación será menor que si estuviera solo.

A continuación se resumen algunos resultados de simulaciones que muestran para tráfico markoviano ON-OFF que alimenta un enlace, cual es la región de aceptación real y la región de aceptación cuando se asume que se mantiene el punto de operación. En el cuadro siguiente se calcula el punto de operación s_i^* en un enlace con 150 fuentes *ONOFF*. En estas fuentes el tiempo ON es 350 ms, y el tiempo OFF es 650 ms. Luego se aumentan las fuentes y se calculan las nuevas pérdidas. Se muestran las pérdidas obtenidas recalculando el punto de operación y las pérdidas estimadas con el punto de operación fijo en s_i^* .

Cantidad fuentes	perdidas exactas(*10 ⁻⁷)	perdidas estimadas(*10 ⁻⁷)
150		
158	2.78	2.97
165	7.23	9.36
172	17	29
180	37	92

Además para las mismas fuentes *ONOFF*, se calculó partiendo de 500 fuentes y una probabilidad de pérdida obtenida de 10⁻⁶ cuantas fuentes adicionales se aceptarían si el

límite de pérdidas aceptable para el enlace fuera 10^{-4} . Se obtuvo en el caso exacto (recalculando el punto de operación) que es posible aceptar 49 fuentes adicionales y en el caso donde se asume el punto de operación fijo (el calculado para las 500 fuentes originales) es posible aceptar 40 fuentes adicionales.

8.5. Capacidad remanente equivalente y Cantidad de fuentes remanentes

Basados en las consideraciones anteriores se introducen en esta sección dos conceptos que se utilizarán en el modelo de ingeniería de tráfico propuesto. Consideremos un LSP que pasa por un conjunto de enlaces L_i y comparte dichos enlaces con otros flujos provenientes de otros LSPs de la red. En cada uno de esos enlaces la función de velocidad de grandes desvíos no podrá superar el logaritmo del ratio de pérdidas de ninguna de las clases que utilizan ese enlace. Por lo tanto, en cada enlace se debe cumplir que si γ_j es menos el logaritmo del ratio de pérdidas de la clase más exigente que utiliza ese enlace entonces:

$$\gamma_j < C_i s_i^* - N s_i^* \sum \rho_k \alpha_k(s_i^*, 1) \quad (8.5.1)$$

siendo $\alpha_k(s^*, 1)$ el ancho de banda efectivo de una fuente del LSP k y ρ_k la proporción de fuentes de dicho LSP sobre el total fuentes del enlace. Se asumirá sin perder generalidad que el LSP 1 es el que interesa analizar. La función de velocidad de grandes desvíos de la probabilidad de pérdida del enlace será

$$N I_i = C_i s_i^* - N s_i^* \sum_{k \neq 1} \rho_k \alpha_k(s_i^*, 1) - N_1 s_i^* \alpha_1(s_i^*, 1) > \gamma_j \quad (8.5.2)$$

con N_1 el número de fuentes que utilizan el LSP 1. Llamaremos capacidad remanente equivalente del LSP 1 en el enlace i a la cantidad de ancho de banda efectivo que puede usar el LSP 1 sin que se viole para ningún LSP en ese enlace los requerimientos de QoS (medidos como ratio de pérdidas). El ancho de banda remanente equivalente dependerá de

la capacidad del enlace, del requerimiento más estricto de QoS de las clases que usan ese enlace, de los restantes LSPs que utilicen el enlace y del punto de operación. La capacidad equivalente remanente que que el LSP 1 puede utilizar en el enlace i (notaremos C_{i1}^{er}) es:

$$C_{i1}^{er} s_i^* = -\gamma_j + C_i s_i^* - N s_i^* \sum_{k \neq 1} \rho_k \alpha_k(s_i^*, 1) \quad (8.5.3)$$

Es importante notar que aún cuando el valor de C_{i1}^{er} en dos enlaces pueda ser el mismo, la cantidad máxima de fuentes del LSP 1 que se pueden hacer pasar por cada enlace puede ser diferente. Esto se debe a que el punto de operación puede ser diferente y por lo tanto a pesar de que la función ancho de banda efectivo de una fuente del LSP 1 es la misma, el valor del ancho de banda efectivo $\alpha_1(s_i^*, 1)$ puede no serlo ya que s_i^* cambia de un enlace a otro.

Introduciremos el concepto de número de fuentes remanentes (N_{1i}^r) que puede usar un LSP en un enlace. El número de fuentes remanentes de la fuente 1 en el enlace i (notaremos N_{1i}^r) depende de C_{i1}^{er} y del valor del ancho de banda efectivo en ese enlace $\alpha_1(s_i^*, 1)$ y es igual a:

$$N_{1i}^r = \frac{C_{i1}^{er}}{\alpha_1(s_i^*, 1)} \quad (8.5.4)$$

Es muy importante notar que para una clase de tráfico se trabajará de aquí en más con la noción de número de fuentes remanentes y no con la de capacidad equivalente remanente. El número de fuentes remanentes dice cuantas fuentes de un LSP pueden pasar por un enlace sin violar las condiciones de QoS. La capacidad equivalente remanente por si sola no brinda una noción de cuanto tráfico de un LSP se puede permitir por un enlace sin violar las condiciones de QoS.

8.6. Ingeniería de tráfico en línea

Resumiendo los resultados de las secciones anteriores, en el modelo más simple que se ha definido para trabajar en línea será necesario calcular y almacenar fuera de línea

- para cada enlace los LSPs que lo atraviesan
- el ancho de banda efectivo de cada clase de fuente que utilizará la red. No es necesario almacenar la función ancho de banda efectivo de cada clase. Cada enlace almacenará su punto de operación y el valor del ancho de banda efectivo de cada clase en su punto de operación.
- el valor medio de cada LSP medido en el ingreso a la red o el valor medio estimado de una 'fuente' de esa clase. En los enrutadores de la frontera, con este valor y el valor medio estimado en línea, se podrá estimar el número de fuentes que atraviesa el enlace en un momento dado.

Con estos datos se puede conocer el ratio de pérdidas de cada LSP al atravesar la red en el momento del inicio de la operación de la red. Si no hay cambios en la topología de los LSPs, se puede conocer el ratio de pérdidas actual midiendo el nuevo valor medio en un instante dado en el ingreso a la red de todos los LSPs. Si hay un cambio topológico de LSPs, se debe calcular en línea qué LSPs utilizan qué enlaces en la nueva configuración y con los nuevos valores medios se puede calcular el ratio de pérdidas de cada LSP en la nueva configuración. Explicaremos más adelante con mayor detalle la implementación de estos puntos.

Cuando se detecta un cambio en el valor medio medido en línea del tráfico de ingreso a la red cada router de la frontera analizará si algún LSP no puede cumplir con sus requerimientos de QoS. Si no se pudieran cumplir todos, se correrán los algoritmos de ingeniería de tráfico que se describen más adelante. Para cada enlace j si el valor medio de cada LSP

i cambia en un cierto coeficiente K_i , el punto de operación calculado fuera de línea era s_j^* y el valor de su ancho de banda efectivo era $\alpha_i(s_j^*, 1)$, entonces el nuevo valor de γ_j será:

$$\gamma_j = C_j s_j^* - \sum_i K_i s_j^* \alpha_i(s_j^*, 1) \quad (8.6.1)$$

Por otra parte cada LSP tendrá un γ_i^{min} para no superar el ratio de pérdida máximo para esa clase de tráfico y por lo tanto ese enlace verificará los requerimientos de QoS si $\gamma_j > \max_i(\gamma_i^{min})$. Para que se cumplan los requerimientos de QoS de toda la red la condición se debe verificar para todos los enlaces. Sobre esta condición y basados en la ecuación 8.6.1 para cada enlace, plantaremos los algoritmos de ingeniería de tráfico que formulamos a continuación.

El orden de los algoritmos que se presentan a continuación, se basa en procurar realizar primero los cambios 'menos costosos' y sólo pasar a los de mayor costo si con los anteriores no se puede asegurar los requerimientos de QoS. Aquí costo está asociado a modificaciones en la configuración de la red. En este sentido es menos costoso modificar los coeficientes de reparto de carga, que establecer un nuevo LSP, ya que en el primer caso se debe modificar sólo los routers de ingreso mientras que en el otro caso se deben modificar además los routers interiores. Por lo tanto el primer algoritmo que se propone se refiere al reparto de carga y el segundo al establecimiento de nuevos LSPs.

El objetivo de los algoritmos presentados será enrutar en línea la carga que arriba a la red MPLS de forma tal que se acepte la mayor cantidad posible de tráfico, sin violar las restricciones de QoS requeridas por cada tipo de tráfico. Por otro lado si no fuera posible enrutar todo el tráfico que desea usar la red, sin violar requerimientos de QoS, se debería notificar al sistema de gestión de que existe necesidad de tomar alguna acción que no puede resolverse mediante los algoritmos en línea propuestos.

En las subsecciones siguientes haremos consideraciones respecto de los algoritmos de reparto de carga y ruteo basado en restricciones propuestos y luego se verá como integrar

estas herramientas en un algoritmo general de TE en línea, los protocolos necesarios y los criterios de cuándo correr dichos algoritmos y cuándo notificar al sistema de gestión.

8.6.1. Reparto de carga en línea

Varios autores han abordado el problema del reparto de carga entre LSPs, que es una de las herramientas que es posible utilizar en la arquitectura MPLS. En estos casos se supone que ya se encuentran configurados un conjunto de LSPs y el problema que se plantea es para cada troncal de tráfico cómo repartir su carga entre los diferentes LSPs asociados a dicha troncal. Algunos autores [20] proponen algoritmos en línea para repartir carga (MATE) y otros [37] han propuesto mecanismos fuera de línea para balancear carga. En ambos casos se busca optimizar alguna medida de performance como por ejemplo las pérdidas promedio o el retardo medio en la red. En el caso de MATE es un algoritmo potente pero asume que el tráfico es constante en media y es muy sensible a las variaciones del parámetro que se debe medir en línea. En la implementación propuesta en ese trabajo se debe medir la derivada del retardo con respecto a la capacidad de cada enlace y para tal fin se debe asumir modelos de los enlaces simples (por ejemplo colas M/M/1). Además no asegura garantías de QoS simplemente optimiza un costo global de la red, como el retardo promedio, pero nada garantiza que para cada agregado se garanticen sus condiciones de QoS. Se puede tener un retardo promedio en la red mínimo y para algunos agregados no cumplir con sus requerimientos de QoS. En este punto interesa señalar que lo que debiera ser un objetivo de un operador no es minimizar el retardo global o las pérdidas, ya que esto no le ofrece por sí mismo ningún beneficio.

Los algoritmos fuera de línea obtienen formas de cálculo que son adecuadas para una optimización de largo plazo de la red, pero que son de difícil aplicación en escalas de tiempo más cortas. Por otra parte la red sufre variaciones de diferente índole y estos algoritmos se basan en estimaciones estadísticas de diversos parámetros en un estado de la red. Este

tipo de algoritmos tratan de optimizar algún parámetro de la red, por ejemplo que las pérdidas máximas en la red sean mínimas, pero como dijimos, esto puede no lograr una configuración donde cada agregado de flujo tenga las pérdidas que se requieren para cumplir sus requerimientos de QoS. Es decir puede convenir que las pérdidas en algunos agregados sean mayores, y aumentar el mínimo global, pero gracias a eso disminuir las pérdidas para el tráfico más restrictivo en cuanto a la QoS.

Existen dos formas de repartir carga entre LSPs:

- por paquete
- por flujo

El primer modo es más simple de implementar y más preciso. El segundo es más complejo de implementar y si no se tienen muchos flujos o estos son muy diferentes es poco preciso. Sin embargo este último presenta una fuerte ventaja que es mantener el ordenamiento de los paquetes de un flujo. Los otros inconvenientes del modelo 'por flujo' pueden ser levantados si se asume un backbone con 'muchos flujos' y con flujos con estadística similar en cada LSP, como es el caso de la propuesta de la tesis. Si bien este modelo es más difícil de implementar, ya existen implementaciones comerciales que así lo hacen [21]. Por las razones antes expuestas se trabajará con el modelo de reparto 'por flujo'. Respecto el modelado matemático del balanceo de carga también es diferente según el modelo utilizado. Llamaremos $N\alpha(s, 1)$ al ancho de banda efectivo de un agregado. En el caso de reparto por paquete, asumiendo un modelo fluido se prueba que si se envía $p\%$ de dicho tráfico por un LSP, su ancho de banda efectivo será $Np\alpha(ps, 1)$. En el caso del reparto por flujo el ancho de banda efectivo es simplemente: $Np\alpha(s, 1)$.

El criterio que utilizaremos para repartir carga es

- que siempre se cumplan todos los requerimientos de QoS de toda la red

- que no se bloquee un enlace de los que utiliza ese LSP de forma que en el futuro se pueda aumentar la carga de este LSP o de otros.

Con este criterio la idea es que:

1. la cantidad de flujos que se envíen por cada LSP sea tal que se verifiquen los requerimientos de QoS para la clase de servicio de dicho LSP y de todos los LSPs que utilizan los mismos enlaces que los LSPs del agregado en consideración.

Consideremos un LSP de los que se usarán para repartir carga (lo llamaremos LSP 1). Para asegurarse que se verificarán todos los requerimientos de QoS, se debe buscar el mínimo número de fuentes remanentes $N_{1i}^{rmin} = \min_i(N_{1i}^r)$ para todos los enlace i que atraviesa el LSP de interés (LSP 1 en el ejemplo). Si se supera este número de fuentes en el LSP 1 entonces algún LSP de la red tendrá un ratio de pérdidas superior a sus requerimientos de QoS.

2. que luego de este reparto de carga se pueda seguir aumentando la carga si se suceden nuevos crecimientos de tráfico en este u otros LSPs. Para esto se pueden adoptar dos criterios para repartir carga que se consideran a continuación. Supongamos que N^{tot} es el número total de fuentes de la troncal estimado a partir de las mediciones realizadas. Los dos posibles criterios de reparto que proponemos son:

- Reparto inversamente proporcional. Simplemente se reparte N^{tot} de manera inversamente proporcional al número de fuentes remanentes mínimo N_j^{rmin} para cada uno de los LSPs j sobre los que se quiere repartir la carga. Esto tiene la ventaja de ser un algoritmo muy simple y que procura volcar más carga por los LSPs más descongestionados y ocupando menos los que ya están más 'congestionados'. Sin embargo esto puede no ser lo más razonable en algunos casos. Por ejemplo si se quieren repartir 1000 fuentes de tipo A. Supongamos que en un LSP ($LSP1$) se pueden aceptar a lo sumo 200 fuentes de tipo A y en otro ($LSP2$) 2000. Lo más razonable en este caso probablemente sería asignar las 1000 fuentes al LSP que puede aceptar 2000 ($LSP2$)

y no usar el *LSP1*. De esta forma, si alguna troncal sólo puede usar un LSP que pase por el enlace crítico del *LSP1* (el que permite 200 fuentes en el ejemplo) tendrá mayor capacidad disponible para futuras ampliaciones de tráfico. El reparto inversamente proporcional lo que haría sin embargo es pasar aproximadamente 90 fuentes por el enlace más recargado (*LSP1*) y 910 por el menos cargado (*LSP2*). De esta forma se dejará en el enlace crítico sólo la capacidad de transportar el equivalente a 110 fuentes de tipo A.

- Una propuesta alternativa es repartir carga con coeficientes tales que se maximice el mínimo número de fuentes remanentes luego del reparto. Es decir para cada LSP a usar se tiene una cantidad de fuentes remanentes en función del coeficiente de reparto:

$$f_j = N_j^{rmin} - p_j N^{tot} \quad (8.6.2)$$

entonces debemos buscar un conjunto de coeficientes que verifiquen

$$\text{maximizar}(\min_j(f_j(p_j))) \quad (8.6.3)$$

con las condiciones,

$$\sum_j p_j = 1 \quad (8.6.4)$$

$$1 \geq p_j \geq 0 \quad \forall j \quad (8.6.5)$$

La solución de esta optimización se obtiene de resolver el siguiente sistema de ecuaciones:

$$N_j^{rmin} - p_j N^{tot} = N_k^{rmin} - p_k N^{tot} \quad \forall j, k \quad (8.6.6)$$

$$\sum_j p_j = 1 \quad (8.6.7)$$

$$1 \geq p_j \geq 0 \quad \forall j \quad (8.6.8)$$

Veremos este punto con un ejemplo. Asumimos que tenemos dos LSPs por los que deseamos repartir N^{tot} fuentes. Cada LSP tiene antes de asignarle carga de esta troncal una cantidad mínima de fuentes remanentes N_1^{rmin} y N_2^{rmin} respectivamente. Entonces los coeficientes p_1 y p_2 que maximizan la mínima cantidad de fuentes remanentes se encuentran de las ecuaciones anteriores y valen:

$$p_1 = \frac{1}{2} + \frac{N_1^{rmin} - N_2^{rmin}}{2N^{tot}} \quad (8.6.9)$$

$$p_2 = \frac{1}{2} + \frac{N_2^{rmin} - N_1^{rmin}}{2N^{tot}} \quad (8.6.10)$$

estas ecuaciones valen si $|N_1^{rmin} - N_2^{rmin}| \geq N^{tot}$; en caso contrario uno de ellos vale 0 y el otro 1 ya que conviene hacer pasar toda la carga por uno de los enlaces. Evidentemente además se debe cumplir que $N_1^{rmin} + N_2^{rmin} \leq N^{tot}$ ya que si no se cumple es necesario agregar algún LSP adicional porque no es posible transportar esa cantidad de fuentes por estos LSPs respetando todos los requerimientos de QoS.

8.6.2. Ruteo basado en restricciones

La idea del ruteo basado en restricciones es como ya mencionamos encontrar un LSP que satisfaga un conjunto de restricciones. Las mismas pueden ser de diferente tipo como por ejemplo:

- de QoS (retardo, pérdidas, etc.)
- administrativas, por ejemplo que un cierto LSP no utilice ciertos enlaces

Como se mencionó antes este problema es NP-completo para la mayoría de los casos de interés. Por lo tanto para su aplicación en línea es necesario buscar algoritmos heurísticos. Estos algoritmos, aunque en algunos casos encuentren una solución sub-óptima, brindan una buena solución a la mayoría de los casos. Se han propuesto muchos algoritmos de este tipo en la literatura. Varios de ellos son muy interesantes pero en general adolecen de deficiencias

por diferentes razones como inestabilidad, complejidad, necesidad de mediciones en línea difíciles de llevar a cabo, etc.

En la práctica los algoritmos que hoy en día se utilizan, son variantes sobre un algoritmo muy simple pero eficiente que es el llamado CSPF (Constrained Shortest Path First). Este algoritmo se basa en el algoritmo SPF. La idea es podar los enlaces que no satisfacen alguna restricción y luego aplica el algoritmo SPF para el árbol podado. Por otra parte la condición de QoS que se utiliza en la práctica en este algoritmo es el ancho de banda. Es decir que el algoritmo poda aquellos enlaces que no tienen un ancho de banda disponible (capacidad ocupable del enlace - capacidad reservada por otros LSPs) superior a un cierto valor requerido para el LSP en cuestión.

Caben mencionar dos aspectos: 1. Este algoritmo, si bien permite resolver diversos casos, para muchas fuentes de tráfico tener ancho de banda asegurado no es garantía de QoS (salvo que se reserve el pico de la fuente en cuyo caso se desperdician recursos). 2. La reservas de ancho de banda se hacen en general en el plano de control, es decir no hay ningún control de que efectivamente se respete el ancho de banda en los datos transmitidos, ni hay necesariamente una reserva de ancho de banda en cada enlace a través de algún mecanismo de scheduling. Se asumirá en la propuesta que la reserva se hace sólo en el plano de control.

Para poder aplicar este algoritmo sin modificaciones importantes es necesario utilizar una métrica de tipo cóncavo para realizar la poda del árbol. El retardo es de tipo aditivo y la probabilidad de pérdida es de multiplicativa, por lo que directamente no los podemos utilizar. Sin embargo vimos que el Ratio de Pérdida definido antes es de tipo cóncavo y por lo tanto podemos utilizarlo en este caso.

La idea básica es modificar este algoritmo usando en lugar de la capacidad requerida y reservable, el número de fuentes remanentes en cada enlace y el número de fuentes que se desea enrutar por ese LSP. Es decir en cada paso del algoritmo se podan los enlaces tales que la cantidad de fuentes requeridas sea mayor que la cantidad de fuentes remanentes.

Para controlar retardo además del ratio de pérdidas debemos realizar una consideración adicional. El peso de cada enlace, no será 1 por defecto sino que será igual a la cota máxima del retardo que sabemos es pequeña por enlace, es decir será igual a: B_i/C_i siendo B_i el tamaño de buffer (pequeño en el modelo utilizado) y C_i la capacidad del enlace. Por lo tanto luego de podar el árbol se buscará aquel camino que tenga el menor retardo máximo y como condición de aceptación del LSP se deberá verificar que el peso total del camino (retardo máximo total) sea inferior a cierto valor de retardo máximo requerido para esa clase.

8.6.3. Otras consideraciones adicionales

En las secciones adicionales dimos las pautas básicas y más importantes para realizar ingeniería de tráfico en línea. En esta sección veremos consideraciones y aspectos complementarios que pueden ser importantes en ciertas aplicaciones.

Además de los criterios básicos del algoritmo CSPF modificado propuesto se pueden introducir otros aspectos. Puede ser importante también podar los enlaces por 'afinidad' como se hace hoy en varias implementaciones de CSPF. Se podrá asignar a cada clase de tráfico un 'color' y lo mismo se podrá hacer para cada enlace. Se podarán aquellos enlaces cuyo color no sea afín al color del LSP que se quiere establecer. Esto permitirá administrativamente evitar que ciertas clases de servicio utilicen ciertas partes de la red. Una razón para utilizar este mecanismo de afinidad, es por ejemplo, que en una parte de la red el ISP hubiera subcontratado el transporte a otro proveedor y para ciertas clases no sea económicamente conveniente 'gastar' ancho de banda de esos enlaces subcontratados. Pero esto tiene sentido en un contexto donde se cobre diferente a diferentes clases de tráfico. Esto es viable en el modelo propuesto porque asegurara a cada clase garantías de calidad de servicio. En este sentido puede ser razonable pensar también que ciertas zonas 'caras' de la red solo sean usadas por tráfico de clases que tienen un margen mayor para el ISP.

Este último punto lleva a realizar algunas consideraciones adicionales. La primera se

deriva de un análisis del modelo DiffServ como arquitectura para ofrecer QoS. Este modelo en su implementación habitual, requiere realizar reconfiguraciones ante cambios del tráfico o de la red, que muchas veces no son simples en los nodos interiores de la red. Estas reconfiguraciones más complejas se deben a que DiffServ habitualmente utiliza en los enlaces interiores mecanismos de scheduling como Weight Fair queueing o de dropping como RED (Random Early Detection). Por esta razón se ha procurado en la tesis evitar la utilización de mecanismos complejos de scheduling y dropping. La propuesta de la tesis, si bien usa conceptos de la arquitectura DiffServ, no incluye en su implementación mecanismos particulares de scheduling y dropping, sino que las colas son FIFO y el descarte de paquetes se produce por el llenado del buffer. En la tesis se asegurará el cumplimiento de los requerimientos de calidad de servicio a través de:

1. Análisis de la red fuera de línea, configuración inicial 'off-line' y análisis de la estadística de las distintas clases de servicio ofrecidas (estimación de la función ancho de banda efectivo de cada clase).
2. En cada LSP enrutar una sola clase de servicio, de forma que midiendo el tráfico medio por cada LSP con la periodicidad adecuada, se puede conocer aproximadamente la función ancho de banda efectivo actual y por lo tanto estimar el número de fuentes de esa clase que están usando ese LSP en un instante dado.
3. Fijar entre actualizaciones el punto de operación de cada enlace.
4. Trabajar con muchas fuentes y buffer pequeño sobre la red 'ficticia' ya mencionada, que asegura QoS sobre la red 'real'. Además como se estudió en la sección 6.5.3, se conoce bajo qué condiciones coinciden exactamente las regiones de aceptación de una y otra.
5. Utilizar mecanismos de balanceo de carga y ruteo basado en restricciones para re-enrutar carga cuando detectamos que la red puede salirse de la región de aceptación. La complejidad de estos mecanismos se encuentra sólo en los routers de la frontera LER.

Además, se propone usar en esta arquitectura mecanismos preemptivos como los descritos en el RFC 2702 para establecer y mantener LSPs. En la propuesta de la tesis se propone el uso de estos mecanismos para darle prioridad para transitar tráfico a ciertas clases frente a otras sobre las que se trabajará con política Best effort. Es decir que si a la hora de establecer un LSP una clase de alta prioridad no encuentra otra alternativa para transportar su tráfico, podrá de manera preemptiva desplazar la reserva (o parte de ella) de algún LSP con menor prioridad. Veremos a continuación algunas consideraciones para el uso de este mecanismo.

Por último se harán a continuación algunas consideraciones que llevarán a cerrar la propuesta. Dentro de los conceptos manejados por Diffserv, está el de acordar con el cliente las garantías que ofrece la red y asegurar estas garantías a un cierto volumen de tráfico mínimo de cada clase que el cliente podrá transportar por la red. Ese tráfico mínimo se asegura en implementaciones tradicionales de DiffServ reservando ancho de banda (configurando en los nodos interiores políticas de scheduling para reservar ancho de banda) y limitando el tráfico del cliente por ejemplo con un token bucket.

En la propuesta sólo haremos reservas para LSPs en el plano de control, no en el de datos. Entonces ¿cómo asegurar mínimos acordados con el cliente?. Se propone que esos mínimos se configuren inicialmente en LSPs en la red y estos tengan máxima prioridad de forma que no puedan ser levantados preemptivamente. Pero si no limitamos el tráfico ¿es posible asegurar los requerimientos de QoS para esos mínimos? Aquí corresponde hacer algunas consideraciones. El objetivo es que el cliente envíe la mayor cantidad posible de tráfico de las clases de alta prioridad (sobre las que se puede cobrar más). La idea en estas clases no debería ser limitar el tráfico que se permite que el cliente envíe a la red. En DiffServ si el tráfico no cumple el 'profile' acordado, el mecanismo de token bucket cambia los paquetes en exceso de clase de scheduling o los cambia de clase de dropping, o eventualmente los descarta. El problema es que sobre cualquiera de estas tres degradaciones que se realizan al

tráfico en exceso, el ISP no podrá cobrar lo mismo que para el tráfico mínimo asegurado.

En nuestra propuesta si es posible quisiéramos garantizar todos los requerimientos de QoS de la red sin poner cotas superiores para ciertas clases. Ahora bien ¿cual sería la situación ideal para un ISP? Se entiende que sería tener toda la red ocupada por tráfico de las clases altas, respetando un mínimo de transporte que le es asegurado a las clases 'best effort'. Cuando se llegara a esta situación, el ISP debe redimensionar o rediseñar la red. El objetivo será por lo tanto tender a esta situación. Usando los algoritmos de ingeniería de tráfico en línea propuestos se procurará tender a esta situación. Como estaremos sub-utilizando la capacidad de la red con los algoritmos propuestos a esta situación no se llegará probablemente si la demanda de tráfico es creciente luego de varios períodos funcionando en línea con los algoritmos propuestos y de varios ajustes fuera de línea cuando los algoritmos que proponemos lleguen a su límite.

A continuación resumimos el algoritmo general de TE que debe correr en los routers de la frontera y en la sección siguiente veremos aspectos relacionados a protocolos e información de control que es necesario transportar entre los nodos de la red para que esta arquitectura funcione.

1. Se mide en cada LSP el tráfico medio que circula por cada troncal (clase, LER origen, LER destino) de tráfico que arriba a cada LER .
2. Si hay variaciones por encima de un umbral, se verifica que el número estimado de fuentes por cada LSP sea menor al número de fuentes remanentes mínimo para cada LSP. Si esto se verifica no se actúa, sólo se ajusta la reserva del LSP al nuevo valor al nuevo valor del número de fuentes.
3. Si el número estimado de fuentes actuales es mayor que el número de fuentes remanentes mínimo para algún LSP, se recalcula el reparto de carga con alguno de los algoritmos de reparto de carga vistos antes.
4. Si el número estimado de fuentes total de la troncal es mayor que el numero de

fuentes remanentes para esa troncal (sumado en todos los LSPs sobre los que reparte carga), entonces se corre el algoritmo ruteo basado en restricciones visto antes tratando de encontrar un nuevo LSP que cumpla:

a) Número de fuentes remanentes mínimo del LSP mayor que el número de fuentes adicionales requeridas por la troncal

b) Criterios de afinidad

c) Retardo mínimo a partir de los pesos de los enlaces como ya se explicó

d) Una vez calculado el LSP retardo total menor del LSP (suma de los pesos de los enlaces) menor que el requerido para esa clase. Una vez establecido el LSP se reparte carga como en el paso

5. Si no se puede encontrar ningún camino para establecer el LSP con el algoritmo 4, se vuelve a correr 4 pero usando para podar el árbol en el algoritmo CSPF la condición de que el número de fuentes adicionales requeridas por la troncal sea menor que: el número de fuentes remanentes mínimo disponible sin contar el tráfico de clases de menor prioridad. Si de esta forma se puede encontrar un camino, se establece preemptivamente y se liberan los LSPs que sean necesarios y que tengan prioridad menor para establecer esta reserva.

Hay que hacer notar que es necesario asegurar mínimos de tráfico a las clases de prioridad menor. Esto se puede lograr estableciendo LSPs para estos mínimos y con prioridad máxima para ser mantenidos una vez establecidos, de forma que no puedan ser levantados. Se levantarán preemptivamente los LSPs de clase más baja establecidos para tráfico adicional al mínimo.

6. Si luego de aplicar los procedimientos anteriores no es posible transportar todo el tráfico solicitado, se establecerá un aviso al sistema de gestión para que reconfigure con otros criterios que los que se puede manejar en línea. La política a adoptar a partir de allí podrá ser no aceptar más conexiones si esto se emplea con algún mecanismo de control de admisión o configurando los token buckets adecuados para llegar al límite del tráfico que

se está transportando actualmente y filtrar tráfico adicional hasta tanto no se reconfigure globalmente la red.

8.6.4. Control distribuido vs centralizado, Protocolos e información que se requiere difundir

La arquitectura planteada en las secciones precedentes requiere una configuración inicial fuera de línea. En esta se establecerán los óptimos de la configuración por ejemplo para los mínimos de tráfico acordado con los clientes. Posteriormente los algoritmos que hemos explicado ajustarán estos óptimos al tráfico real hasta que detecten que se llegó al límite de la región de aceptación que es posible obtener con estos algoritmos. Allí notificarán al sistema de gestión para que tome las acciones correspondientes.

Ahora bien, en la etapa en línea los algoritmos pueden ser ejecutados de manera distribuida sin necesidad de intervención de los operadores o de un control de gestión centralizado. Para esto es necesario:

1. En los routers de la frontera ejecutar los algoritmos explicados antes.
2. Cuando se detecta que es necesario establecer un nuevo LSP o modificar una reserva de recursos de uno existente, será necesario un algoritmo de distribución de etiquetas con posibilidad de ruteo explícito, esto puede ser hecho con RSVP o CR-LDP tal como se hace hoy. La diferencia es que en la reserva se transmitirá el número de fuentes del LSP. Este número junto con el ancho de banda efectivo en cada enlace de una fuente de la clase a la que pertenece el tráfico del LSP, permiten calcular la reserva de ancho de banda del LSP en cada enlace.
3. Un protocolo que sea capaz de transportar información del estado de los enlaces. Esto puede hacerse por ejemplo con las modificaciones existentes hoy al protocolo OSPF (OSPF-TE [42]) y que tiene ya implementaciones comerciales. La diferencia es que la información transportada no es la misma que con los algoritmos actuales de CSPF. En estos se transporta al menos: el peso de los enlaces, la capacidad reservable del enlace, la capacidad reservada y

el 'color' del enlace. Para la propuesta de la tesis es necesario transportar al menos: el peso de los enlaces (relacionado con el retardo como se explicó antes), la capacidad reservable del enlace (C_i), el ancho de banda efectivo de una fuente de cada clase que utilice la red en el punto de operación del enlace, el número de fuentes reservadas de cada clase en ese enlace, el punto de operación en que funciona el enlace s_i^* y el 'color' del enlace. Con esta información de estado es posible correr los algoritmos anteriores. En los LER de la frontera es necesario conocer además el valor requerido de QoS (en ratio de pérdida) para cada clase de servicio.

8.7. Resultados numéricos

En la figura se muestra la arquitectura del backbone que utilizaremos para este estudio numérico.

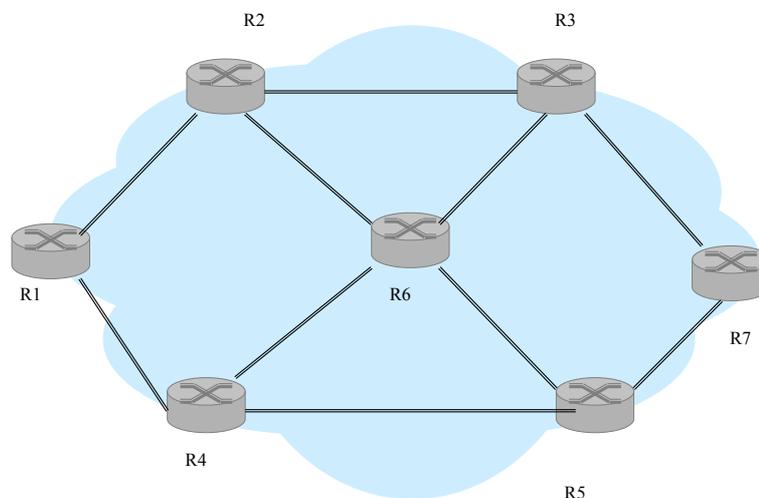


Figura 8.2: Arquitectura del backbone del caso de estudio

Asumimos que existen dos clases de tráfico que interesan a los efectos de este ejemplo (clase A, y clase B). La clase A la modelaremos con un tráfico fluido markovino ON-OFF con

los valores sugeridos en [11] para modelar el tráfico de voz. Se modelará con $h_{ON} = 64kb/s$, $T_{ON} = 352ms$ y $T_{OFF} = 650ms$. El tráfico clase B se modeló con una fuente markoviana ON-OFF con $T_{ON} = T_{OFF}$ y $h_{ON} = 64kb/s$. A los efectos del ejemplo asumiremos que el ratio de pérdidas L , admitido para cada clase es $L_{claseA} = 10^{-6}$ y $L_{claseB} = 10^{-4}$. Asumimos por simplicidad en el ejemplo que todos enlaces del backbone son de 100Mb/s. En la figura 8.3 se muestra la topología de LSPs establecidos por el operador fuera de línea, habiendo medido para realizar esta configuración las trazas de cada tráfico por cada LSP configurado. Con estas trazas se estimó la función ancho de banda efectivo y se estimó la cantidad de fuentes que utilizan en este régimen cada LSP. Esta estimación de cantidad de fuentes que utilizan cada LSP se muestra en la figura 8.4. Resumimos las características de cada LSP con la configuración y estimaciones fuera de línea.

LSP1: clase A; enlaces: R1-2, R2-3, R3-7; 1000 fuentes estimadas

LSP2: clase A; enlaces: R1-4, R4-5, R5-7; 1000 fuentes estimadas

LSP3: clase B; enlaces: R1-2; 1000 fuentes estimadas

LSP4: clase B; enlaces: R1-4; 1000 fuentes estimadas

LSP5: clase B; enlaces: R2-3; 2000 fuentes estimadas

LSP6: clase B; enlaces: R4-5; 2000 fuentes estimadas

LSP7:clase B; enlaces: R3-7; 1000 fuentes estimadas

LSP8:clase B; enlaces: R5-7; 1000 fuentes estimadas

LSP9: clase B;enlaces: R2-5,R6-5; 2200 fuentes estimadas

LSP10: clase A; enlaces:R2-5,R6-5; 700 fuentes estimadas

LSP11: clase B; enlaces: R4-6,R6-3; 2200 fuentes estimadas

LSP12: clase A;enlaces: R4-6,R6-3; 700 fuentes estimadas

Con esta configuración además se calculan los puntos de operación de cada enlace. El valor de s^* y del ancho de banda efectivo de cada clase de fuente($ebwA$ y $ebwB$) en cada enlace se lista a continuación. Se listan también a continuación para cada enlace la utilización

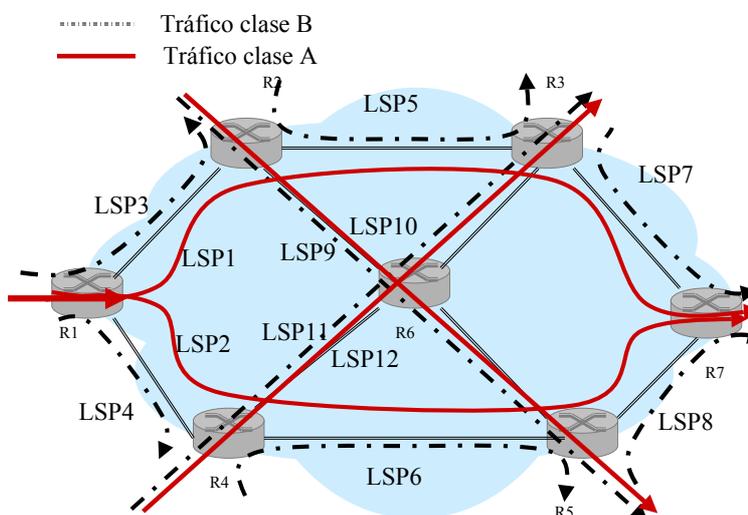


Figura 8.3: Topología de LSPs establecidos en el backbone

del mismo con esta configuración fuera de línea. Se muestra también para cada enlace, el número de fuentes adicionales (a las que ya utilizan el enlace) que se podrían aceptar del tráfico de clase A cumpliendo con los requerimientos de QoS con esta configuración.

Enlace	s^* (1/Mbits)	ebwA(kb/s)	ebwB(kb/s)	utilización(%)	Fuentes adicionales A
1-2	25.2	34.9	43.7	54.5	603
2-3	4.56	24.7	34.3	86.5	171
3-7	25.2	34.9	43.7	54.5	603
1-4	25.2	34.9	43.7	54.5	603
4-5	4.56	24.7	34.3	86.5	171
5-7	25.2	34.9	43.7	54.5	603
2-6	4.8	24.8	34.4	86.1	190
6-5	4.8	24.8	34.4	86.1	190
4-6	4.8	24.8	34.4	86.1	190
6-3	4.8	24.8	34.4	86.1	190

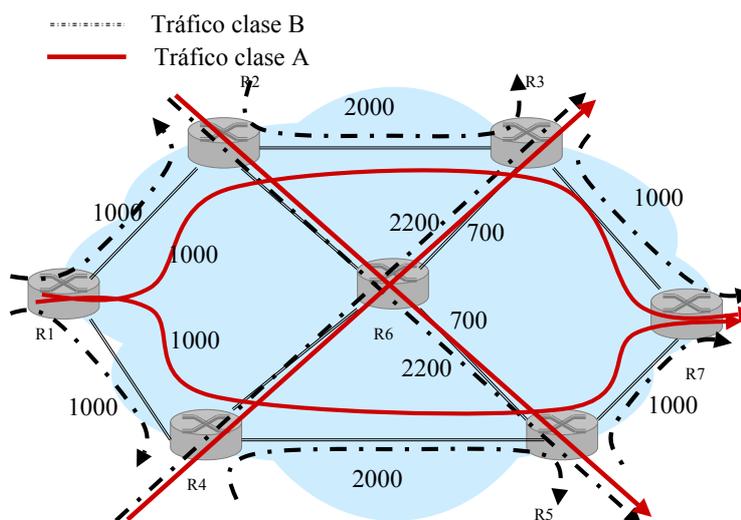


Figura 8.4: Cantidad de fuentes por cada LSP estimada fuera de línea

Como vemos se está balanceando carga del tráfico de clase A que ingresa por el router R1 entre los LSPs: LSP1 y LSP2 con coeficientes de reparto de 50% por cada uno.

Supongamos que la red comienza a operar con esta configuración y en determinado

momento midiendo el valor medio del tráfico por el LSP6 se detecta una reducción de un 7% aproximadamente del tráfico. Esto lleva el número estimado de fuentes de 2000 a 1850 en este LSP. Ante este cambio lo único que ocurre es que a través del protocolo de establecimiento de LSPs utilizado (RSVP-TE por ejemplo) se cambia la reserva en el plano de control para los enlaces que utiliza este LSP para ajustar el nuevo ancho de banda efectivo. En particular este tráfico del backbone en estudio utiliza solo el enlace 4-5, por lo que se ajusta la reserva del ancho de banda efectivo en ese enlace para el LSP6 de: $2000 * 34,9(kb/s)$ a $1850 * 34,9(kb/s)$.

Esto cambia la cantidad de fuentes remanentes de clase A. Antes de este cambio se tenía la siguiente situación:

$$s^* \alpha_A(s^*, 1) = 4,56 * 24,7 * 10^{-3} = 0,1126$$

$$N_{LSP6} * s^* \alpha_B(s^*, 1) = 4,56 * 2000 * 34,3 * 10^{-3} = 312,8$$

$$\text{El ratio de pérdidas admitido es: } \log_e(10^{-6}) = -13,8$$

usando Bahadur-Rao (ver sección 5.3):

$$-13,8 = -NI_A - (0,5 * \log(-NI_A * 4 * \pi))$$

entonces,

$$NI_A = 11,3$$

$$s^* C_{4-5} = 4,56 * 100 = 456$$

$$N^{\text{remanentes}A} = (-11,3 + 456 - 312,8)/0,113 = 131,884/0,1126 = 1171 \text{ fuentes}$$

Como por el LSP2 circulaban 1000 fuentes de tráfico de clase A, la cantidad de fuentes adicionales era 171.

Al cambiar la reserva de 2000 fuentes a 1850 en el tráfico del LSP6, la cantidad de fuentes remanentes del tráfico de clase A en el enlace 4-5 cambia de la siguiente forma:

$$N_{LSP6} * s^* \alpha_B(s^*, 1) = 4,56 * 1850 * 34,3 * 10^{-3} = 289,35$$

$$N^{\text{remanentes}A} = (-11,3 + 456 - 289,35)/0,113 = 155,35/0,1126 = 1380 \text{ fuentes.}$$

Por lo tanto la cantidad de fuentes adicionales de tráfico de clase A que se podrán hacer circular

en esta nueva situación será de 380 fuentes.

Asumamos que posteriormente se incrementa el tráfico de clase A que ingresa por el router 1 en un 10 %. Esto implica que se deban transportar por los LSP1 y LSP2, 100 fuentes más por cada uno. Esto lleva a un total de 1100 fuentes por cada LSP, lo que es compatible con la cantidad de fuentes remanentes para cada LSP. Esto se debe a que el enlace con menor cantidad de fuentes remanentes adicionales para el LSP1 es el 2-3 y puede transportar 171 fuentes. Para el LSP2 es el 4-5 que luego del cambio de tráfico visto antes puede transportar 380 fuentes más.

Ante esta situación, en la red no se toma ninguna acción, excepto que el router 1 a través de RSVP-TE por ejemplo, ajusta en el camino de cada LSP la reserva a 1100 fuentes.

Luego se vuelve a incrementar el tráfico de clase A que ingresa por el router 1 en otro 10 %. Con los coeficientes de reparto de carga actuales eso requiere que por cada LSP se transporten 1210 fuentes. El router1 detecta que por el LSP1 no es posible pasar 1210 fuentes de este tráfico porque el número de fuentes remanentes es 1171. En el LSP2 en cambio no hay inconveniente ya que el número de fuentes remanentes es 1380. Se observa también que en total se requieren 2420 fuentes y entre ambos LSPs se pueden transportar $1380 + 1171 = 2551$. Por lo tanto se resuelve ajustar los coeficientes de reparto de carga según el criterio establecido en 8.6.9. De estas ecuaciones los coeficientes de reparto de carga quedan:

$$p1 = 0.5 + (1171-1380)/(2*2420) = 0.457$$

$$p2 = 0.5 + (1380-1171)/(2*2420) = 0.543$$

$$\text{Cantidad de fuentes LSP1} = 0.457 * 2420 = 1105$$

$$\text{Cantidad de fuentes LSP2} = 0.543 * 2420 = 1315$$

Suponemos ahora que se produce un nuevo aumento de un 10 % de la troncal que se enruta por los LSPs LSP1 y LSP2. Esto lleva el número estimado de fuentes a 2662. Con los dos LSPs establecidos no es posible transportar esta cantidad de fuentes. Por lo tanto se

corre un algoritmo de CSPF modificado según se explicó. Es necesario establecer un nuevo LSP que tenga un número de fuentes remanentes mínimo para el tráfico de clase A de al menos: $2662-2551 = 111$ fuentes de tráfico de clase A adicionales. Si se podan los enlaces que no satisfacen estos requerimientos. A continuación resumimos la cantidad de fuentes remanentes de tráfico de clase A que puede transportar cada enlace, en la configuración actual:

Enlace	s^* (1/Mb)	ebwA(kb/s)	ebwB(kb/s)	Fuentes adicionales A
1-2	25.2	34.9	43.7	498
2-3	4.56	24.7	34.3	66
3-7	25.2	34.9	43.7	498
1-4	25.2	34.9	43.7	288
4-5	4.56	24.7	34.3	65
5-7	25.2	34.9	43.7	288
2-6	4.8	24.8	34.4	190
6-5	4.8	24.8	34.4	190
4-6	4.8	24.8	34.4	190
6-3	4.8	24.8	34.4	190

Los caminos posibles son:

- 1-2,2-6,6-5,5-7. En este caso el número de fuentes remanentes mínimo para el LSP es 190 fuentes.
- 1-2,2-6,6-3,3-7. En este caso el número de fuentes remanentes mínimo para el LSP es 190 fuentes.
- 1-4,4-6,6-5,5-7. En este caso el número de fuentes remanentes mínimo para el LSP es 190 fuentes.

- 1-4,4-6,6-3,3-7. En este caso el número de fuentes remanentes mínimo para el LSP es 190 fuentes.

Los demás caminos posibles no cumplen con el requerimiento de fuentes mínimas de tráfico de clase A.

Cómo además todos los caminos tienen el mismo costo en cuanto al retardo (iguales capacidades y tamaño de buffer en todos los enlaces), y todos tienen el mismo número de fuentes remanentes mínimo, se elige uno cualquiera de ellos, por ejemplo el 1-2,2-6,6-5,5-7.

Ahora se dispone de 3 LSPs para repartir carga. Se deben repartir 2662 fuentes, por lo cual se calculan los nuevos coeficientes de reparto:

$$p1 = 0.333 + (2*1171-1380-190)/(3*2662) = 0.43$$

$$p2 = 0.333 + (2*1380-1171-190)/(3*2662) = 0.508$$

$$p3 = 0.333 + (2*190-1171-1380)/(3*2662) = 0.062$$

$$N1 = 1145 \quad N2 = 1352 \quad N3 = 165$$

De esta forma la utilización y las fuentes remanentes de cada clase de los diferentes enlaces queda

Enlace	s^* (1/Mb)	ebwA(kb/s)	ebwB(kb/s)	utilización	Fuentes adicionales A
1-2	25.2	34.9	43.7	61.5	293
2-3	4.56	24.7	34.3	89.7	26
3-7	25.2	34.9	43.7	57.7	458
1-4	25.2	34.9	43.7	62.4	251
4-5	4.56	24.7	34.3	89.6	28
5-7	25.2	34.9	43.7	66.1	86
2-6	4.8	24.8	34.4	89.9	25
6-5	4.8	24.8	34.4	89.9	25
4-6	4.8	24.8	34.4	86.1	190
6-3	4.8	24.8	34.4	86.1	190

Por último respetando todos los requerimientos de ratio de pérdidas y sin modificar los puntos de operación calculados fuera de línea se estudió cual es la máxima utilización que podemos lograr en cada enlace aumentando lo máximo posible el tráfico de clase A y luego en aquello que aún se podía aumentando al límite el tráfico de clase B. Los resultados de utilización se muestran en la siguiente tabla:

Enlace	s^* (1/Mbytes)	ebwA(kb/s)	ebwB(kb/s)	utilización	Fuentes adicionales A
1-2	25.2	34.9	43.7	69	0
2-3	4.56	24.7	34.3	91	0
3-7	25.2	34.9	43.7	69	0
1-4	25.2	34.9	43.7	69	0
4-5	4.56	24.7	34.3	91	0
5-7	25.2	34.9	43.7	69	0
2-6	4.8	24.8	34.4	91	0
6-5	4.8	24.8	34.4	91	0
4-6	4.8	24.8	34.4	91	0
6-3	4.8	24.8	34.4	91	0

Como se puede ver en varios enlaces se logra una utilización alta asegurando la calidad de servicio de las diferentes clases de tráfico. En algunos enlaces se alcanza una utilización no muy alta $\approx 70\%$. Estos enlaces son precisamente aquellos en que fuera de línea cuando se calculó el punto de operación se tenía una utilización muy baja $\approx 50\%$. Evidentemente el punto de operación es muy alto en estos casos y la variación al aumentar el tráfico es muy grande. Por ejemplo para el enlace 1-2, si se re-calcula el punto de operación en la nueva situación se obtiene $s^* = 12,5$ es decir se redujo prácticamente a la mitad y con ese nuevo valor del punto de operación, las pérdidas son en realidad despreciables, mucho menores que las calculadas con el punto de operación anterior. Esto lleva a pensar en la posibilidad de

usar un punto de operación menor que el calculado para aquellos casos en que la utilización obtenida fuera de línea es muy baja.

8.8. Fortalezas y debilidades de la arquitectura propuesta, posibles mejoras

La arquitectura permite asegurar QoS y ajustar de manera automática la configuración de la red para hacer un mejor uso de recursos asegurando las condiciones de QoS. No requiere sustanciales modificaciones a los protocolos propuestos y en operación hoy en los enrutadores comerciales. Requiere sí el cálculo fuera de línea del ancho de banda efectivo de las diferentes clases transportadas por la red y una configuración inicial que asegure requerimientos mínimos de tráfico que se debe soportar para cumplir los compromisos con los clientes.

Los algoritmos de CSPF actuales corren a pedido del operador que solicita calcular un nuevo LSP con cierto ancho de banda. En la propuesta de la tesis corre automáticamente cuando se detecta que es necesario para cumplir con los requerimientos de QoS. No es necesaria la intervención del operador.

Integra también, el balanceo de carga para minimizar el establecimiento de LSPs si es posible utilizar LSPs ya existentes. La configuración de los nodos interiores es simple, no existiendo necesidad de configuraciones complejas en cuanto a políticas de scheduling o dropping. Se procura ocupar la red con aquellos tipos de tráfico que generan mayores beneficios al operador respetando mínimos para las clases más bajas.

Sin embargo la arquitectura tiene una debilidad. Básicamente la restricción en ratio de pérdidas para un LSP en un enlace viene dada por la clase más restrictiva con la que comparte dicho enlace el LSP. Para entender este punto pensemos en un caso extremo en el cual dos LSPs uno de una clase alta y otro de una clase baja comparten todos los enlaces. En ese caso, a la clase baja le estaremos asegurando la misma QoS que a la clase alta.

Esto dependiendo de la topología de la red puede implicar una posible sub-optimización de recursos. Esto se debe a al afán por simplificar la gestión de las colas en los enlaces y usar una arquitectura muy simple haciendo compartir a todas las clases una misma cola. Esto representa un beneficio ya que simplifica la gestión de los routers interiores, siendo por tanto más escalable la red.

Si este punto representa un problema, el mismo se puede atenuar por dos vías: 1. Dividir el ancho de banda de los enlaces de la red por donde se pueda transportar clases con requerimientos fuertes de QoS y clases de tipo best effort. El tráfico 'best effort' utilizará cierto ancho de banda mínimo reservado para él y el resto se utilizará por las clases con requerimientos más fuertes. Todo lo expuesto en las secciones anteriores se aplica tal como se describió sobre la parte de la capacidad reservada para las clases con requerimientos más exigentes. 2. Procurando separar zonas de la red en que se dé preferencia al tráfico best effort, de zonas por donde se transporte con preferencia el tráfico con requerimientos fuertes de QoS. Con mecanismos de afinidad como los explicados antes se puede resolver este problema.

Por otra parte, estos dos mecanismos no tienen por qué ser excluyentes.

Otra crítica que se puede hacer a la arquitectura propuesta es que se reserva un valor de ancho de banda efectivo que no es el valor que debiera reservarse ya que el punto de operación de un enlace cambia al cambiar el tráfico que por él circula. Como dijimos el usar un punto de operación que no es el óptimo sobreestima las pérdidas, por lo cual estamos seguros que se cumplen los requerimientos de QoS. Pero en la actualidad con los algoritmos de CSPF la forma de reservar capacidad para un LSP es a través del valor de pico si se quiere asegurar QoS. La arquitectura propuesta sobreestima la QoS, pero se sabe que el valor de ancho de banda efectivo está entre el pico y la media del tráfico, por lo cual se sobreestima menos que con los mecanismos actuales.

Otra crítica es que al utilizar la teoría de grandes desvíos estamos teniendo una aproximación asintótica logarítmica. Esto lleva a que se tenga una aproximación que en muchos casos sobreestima mucho las pérdidas. Este problema ha sido abordado por varios autores buscando asintóticas más ajustadas. Una de ellas ya la mencionamos y brinda una asintótica más ajustada basada en el teorema de Bahndur - Rao. Recientemente Mazumdar et al. [34] han propuesto una asintótica que da una mayor precisión aún que la anteriormente mencionada.

8.9. Conclusiones

En este capítulo basándonos en el estudio y análisis realizado en los capítulos precedentes, se ha propuesto una arquitectura para realizar ingeniería de tráfico en línea en una red MPLS y asegurar requerimientos de QoS.

Esta arquitectura se basa en modificar el algoritmo CSPF y en utilizar balanceo de carga. Con estas dos herramientas se plantea una solución que permite asegurar QoS a los agregados de flujo.

Esta solución no es óptima, en el sentido que puede ser posible aumentar la carga en la red sobre los máximos calculados y aún cumplir los requerimientos de QoS. Sin embargo, estamos seguros que si aceptamos una situación los requerimientos de QoS se verificarán. Se vio también en un ejemplo numérico que se pueden obtener valores elevados de utilización de los enlaces de la red utilizando los algoritmos propuestos.

Capítulo 9

Conclusiones y trabajo futuro

A lo largo de la tesis hemos estudiado el estado del arte en los tres temas propuestos:

- Ingeniería de Tráfico en MPLS y su relación con QoS en Internet
- Grandes Desvíos aplicados al análisis de redes de Telecomunicaciones
- Estimación de parámetros de QoS en una red.

En la tesis resumimos el estado del arte en estas tres áreas y en cada una de ellas realizamos hemos procurado realizar un aporte:

- Ingeniería de Tráfico en MPLS y su relación con QoS en Internet: Hemos propuesto una arquitectura que permite realizar balanceo de carga y ruteo basado en restricciones en línea en una red MPLS.
- Grandes Desvíos aplicados al análisis de redes de Telecomunicaciones: Hemos profundizado en el estudio de performance de una red MPLS utilizando una 'red ficticia' equivalente a la red real. Realizamos un análisis de cuando es posible si cometer error utilizar los resultados obtenidos usando la red ficticia en la red real. También hemos visto de qué depende la magnitud del error cometido cuando el análisis no es exacto y el impacto de este error en el análisis de punta a punta de una red.

- Estimación de parámetros de QoS en una red: Hemos propuesto una metodología para ajustar el estimador de ancho de banda efectivo para una traza de tráfico de una fuente con un modelo markoviano fluido. Hemos también encontrado junto con otros integrantes del grupo ARTES, bajo qué condiciones el estimador del punto de operación de un enlace y la probabilidad de pérdida del mismo es consistente y cuándo es válido un TCL para dichos estimadores.

El tema abordado es muy amplio y en el desarrollo de la tesis han ido apareciendo diversos aspectos que pueden ser mejorados o analizados con mayor detalle. Algunos de estos puntos son:

- Ingeniería de Tráfico en MPLS y su relación con QoS en Internet: La propuesta se basa en modificar el algoritmo CSPF y en proponer dos algoritmos de reparto de carga en línea. Con el mismo marco teórico se pueden pensar otros algoritmos diferentes para realizar balanceo de carga y CBR diferentes a los aquí propuestos. Una comparación detallada de ventajas y desventajas de diversos algoritmos y su implementación en una red real serían puntos que contribuirían a aclarar y probablemente a mejorar la propuesta.
- Grandes Desvíos aplicados al análisis de redes de Telecomunicaciones. En este punto hemos visto que hay aún pocos resultados relativos al análisis de performance de punta a punta de una red. El resultado que se utilizó es válido en un contexto del régimen asintótico de muchas fuentes y buffer pequeño. Aquí se abre toda un área de desarrollo entorno al modelado y análisis de performance de redes y en particular para el estudio de performance de punta a punta de una red.
- Estimación de parámetros de QoS en una red. En esta área existen diversos aspectos que aún quedan abiertos. ¿Es posible estimar la fórmula 'infsup' de forma eficiente para que sea aplicada en línea? ¿es posible establecer alguna fórmula plug-in para

el intervalo de confianza de la probabilidad de pérdida, el retardo u otros parámetros de QoS? ¿existen otros índices de performance que brinden mejores resultados para estimar la calidad de servicio de punta a punta?

Los puntos anteriores son solo algunos ejemplos de los temas que deberán ser encarados como trabajo futuro a los efectos de mejorar la propuesta.

Apéndice A

Apéndice: Software desarrollado

A.1. Introducción

Para realizar las simulaciones y análisis numéricos que se explicaron a lo largo de la tesis fue necesario utilizar y en algún caso desarrollar diversas herramientas. Muchos de los resultados se realizaron mediante pequeños programas en Matlab. Lo que es este apéndice explicaremos brevemente no son esos programas en Matlab sino una herramienta desarrollada en Java y que permite realizar diversas simulaciones y estudios de trazas de tráfico y de la performance de un enlace en el régimen asintótico de muchas fuentes.

A.2. Principales funcionalidades del software

El software desarrollado en JAVA presenta tres funcionalidades básicas:

- Generación de trazas simuladas de tráfico a partir de un modelo. Se aceptan diferentes modelos (Poisson, Markoviano discreto, markoviano continuo, gaussiano). Para esto se especifican los parámetros del modelo y se generan trazas simuladas con ese modelo.
- Estimación del ancho de banda efectivo. Para esto se estima el ancho de banda efectivo de una traza de tráfico por tres mecanismos: estimador temporal no paramétrico, estimador poisson, estimador markoviano continuo y estimador markoviano discreto. Para esto se ingresa un rango de los parámetros espacial y temporal en los que se

quiere calcular función ancho de banda efectivo y se calcula l matriz $\alpha(s, t)$ en esos rangos.

- Cálculo de los parámetros de QoS. Permite resolver la fórmula insup y calcular la probabilidad de pérdida. Para esto se debe ingresar la combinación de fuentes de tráfico que ingresan al enlace (indicando sus funciones ancho de banda efectivo), la capacidad y el tamaño de buffer del enlace. Permite también ajustar la estimación por el método de Bahadur-Rao.

A.3. Interfaz de usuario

En las siguientes figuras se muestra la interfaz de usuario de la herramienta desarrollada.

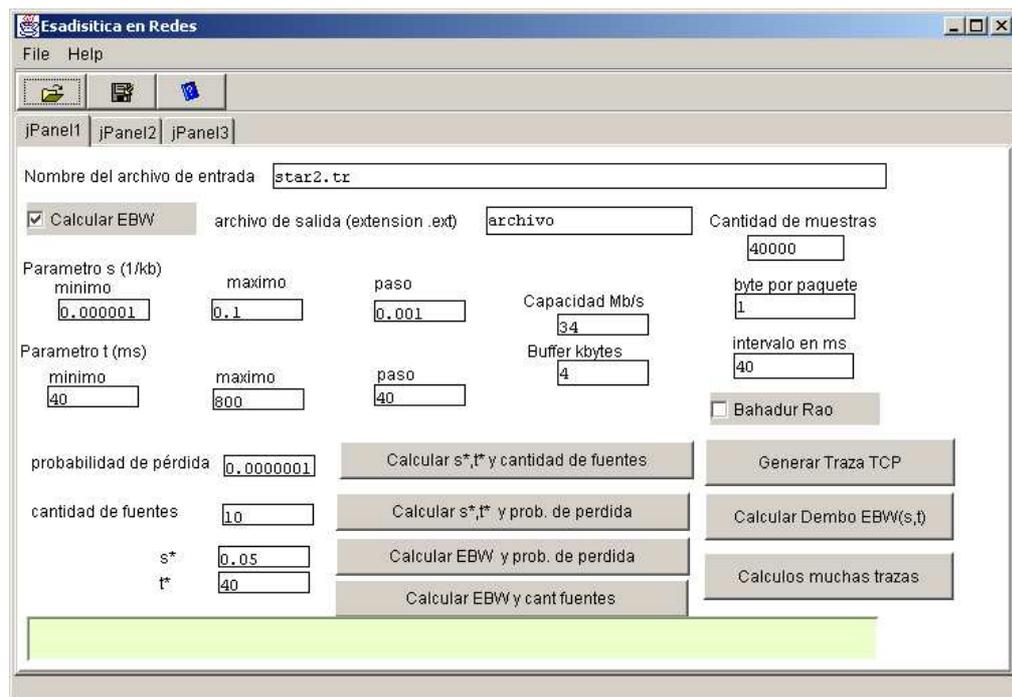


Figura A.1: ventana de usuario del soft desarrollado

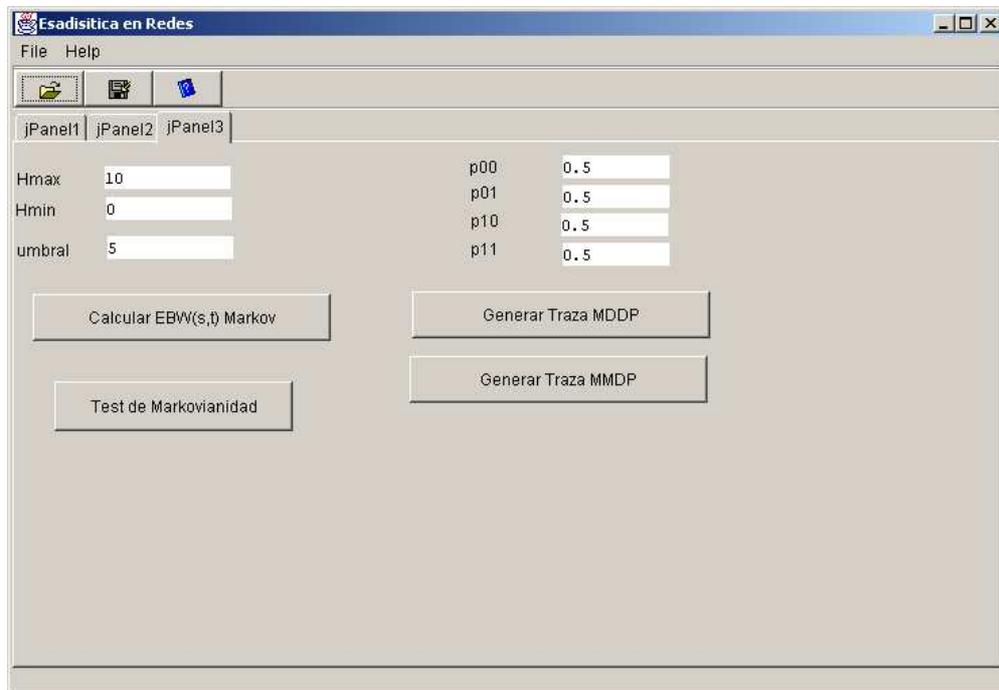


Figura A.2: ventana de usuario del soft desarrollado

En la primera ventana se permite ingresar un archivo con una traza y calcular el estimador temporal no paramétrico del ancho de banda efectivo. También se permite ingresando el nombre del archivo donde se tiene la estimación del ancho de banda efectivo de una traza, la capacidad del enlace y el tamaño del buffer calcular:

1. El punto de operación y el número de fuentes que alimentan el enlace resolviendo la fórmula infsup , a partir del ingreso de una probabilidad de pérdida que se desea para enlace.
2. El punto de operación y la probabilidad de pérdida del enlace para un número de fuentes dado.
3. Dado un punto (s,t) y la cantidad de fuentes, el ancho de banda efectivo en ese punto y la probabilidad de pérdida
4. Dado un punto (s,t) y una probabilidad de pérdida deseada, el ancho de banda efectivo en ese punto y la cantidad de fuentes.

También se puede ajustar los valores usando Bahadur-Rao si se desea.

En la segunda ventana se permite por un lado calcular dado un valor de umbral y los valores de la velocidad de transmisión de la fuente, el ancho de banda efectivo con un modelo markoviano ON-OFF. Se permite también correr un test de markovianidad sobre la traza.

En esa venta además se permite generar trazas con un modelo markoviano discreto (MDDP) y con un modelo markoviano continuo(MMDP).

A.4. Clases y documentación del software

Por el volumen de esta información no se incluyó en el informe, pero se encuentra disponible vía Web en: *<http://iie.fing.edu.uy/~belza/doc/>*

Bibliografía

- [1] David Allan, *Guidelines for MPLS load balancing, draft-allan-mpls-loadbal-04*, IETF (April 2003).
- [2] ARTES L. Aspirot, P. Belzarena, P. Bermolen, A. Ferragult, G. Perera, and M. Simon, *Quality of service parameters and link operating point estimation based on effective bandwidths*, HET-NET03 (2003).
- [3] D. Awduche and J. Malcolm, *Requirements for traffic engineering over MPLS RFC2702*, IETF (1999).
- [4] F. Baccelli and P. Bremaud, *Elements of queueing theory - palm martingale calculus and stochastic recurrences*, Springer-Verlag, 1994.
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *an architecture for Differentiated Services RFC 2475*, IETF (Dec. 1998.).
- [6] JEAN-YVES LE BOUDEC and PATRICK THIRAN, *NETWORK CALCULUS a theory of deterministic queueing systems for the internet*, Springer Verlag - LNCS 2050,, 2002.
- [7] R. Braden, D. Clark, and S Shenker, *Integrated services in the internet architecture: An overview RFC 1633*, IETF (Jun. 1994).
- [8] Ramon Casellas, *MPLS traffic engineering*, Ph.D. thesis, ENST, Paris, 2002.
- [9] Cheng-Shang Chang, *Stability, queue length and delay of deterministic and stochastic queueing networks*, IEEE Transactions on Automatic Control (1994), 39:913–931.

- [10] C.S. Chang, *Performance guarantees in communication networks*, Springer Verlag, TNCS, IV, 1, 2000.
- [11] Courcoubetis and Siris, *Procedures and tools for analysis of network traffic measurements*, Elsevier Science (2001).
- [12] C. Courcoubetis and R. Weber, *Buffer overflow asymptotics for a switch handling many traffic sources*, Journal of Applied Probability (1996).
- [13] Cruz, *A calculus for network delay, part i: Network elements in isolation.*, IEEE Trans. Inform. Theory, vol 37-1 (January 1991.), 114–131.
- [14] ———, *A calculus for network delay, part ii: Network analysis.*, IEEE Trans. Inform. Theory, vol 37-1 (January 1991.), 132–141.
- [15] D. Dacuna-Castelle and M. Dufflo, *Probabilités et statistiques tome*, Masson, Paris, 1993.
- [16] J Daigle, *Queueing theory for telecommunications*, Adison -Wesley, 1991.
- [17] Amir Dembo and Tim Zajic, *Large deviations: From empirical mean and measure to partial sums proces*, Stochastic Process Applications 57 (1995), 191–224.
- [18] Amir Dembo and Ofer Zeitouni, *Large deviations techniques and applications*, 1993.
- [19] N.G. Duffield and Neil O’Connell, *Large deviations and overflow probabilities for the general single server queue, with applications.*, Math. Proc. Cam. Phil. Soc., 118 (1995), 363–374.
- [20] A. Elwalid, C. Jin, H. Low, and I. Widjaja, *MATE: MPLS adaptive traffic engineering*, Proc. IEEE INFOCOM 2001 (2001), 1300–1309.
- [21] E. Osborne and Ajay Simha, *Traffic engineering with MPLS*, Cisco Press, 2003.
- [22] D. Y. Eun and N. Shroff, *Network decomposition in the many-sources regime.*, Advances in Applied Probability (submitted, 2002.).

- [23] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen ., *Multi-protocol label switching (MPLS) support of differentiated services - RFC 3270*, IETF (Mayo 2002).
- [24] A.J. Ganesh and N. O'Connell, *The linear geodesic property is not generally preserved by a fifo queue*, Annals of Applied Probability (1998), 8(1): 98 – 111.
- [25] A. J. Gansh and N. O'Connell, *A large deviation principle with queueing applications*, Stochastic and Stochastic Reports (2002).
- [26] G.Kesidis, J.Walrand, and C.S. Chang, *Effective bandwidth for multiclass markov fluid and other ATM sources*, IEEE ACM Transaction on Networking (1993), 1 (4) 424–428.
- [27] JUNIPER, *MPLS applications*, [Http://Www.Juniper.Net/Techpubs/Software/Junos/Junos56/Swconfig56-Mpls-Apps/ Download/ Swconfig56-Mpls-Apps.Pdf](http://www.juniper.net/techpubs/software/junos/junos56/swconfig56-mpls-apps/download/swconfig56-mpls-apps.pdf).
- [28] F.P. Kelly, *Notes on effective bandwidth*, Stochastic Networks: Theory and Applications, Oxford University Press (1996).
- [29] L. Kleinrock, *Queueing systems vol. i: Theory*, Wiley, New York, 1975.
- [30] H. Korezlioglu, D.Kofman, S. Thome, and L. Decreusefond, *Eléments de théorie des files d'attente*, Ecole Nationale Superieure des Telecommunications, 2001/2002.
- [31] F.A. Kuipers, T. Korkmaz, and M. Krunz, *An overview of constraint-based path selection algorithms for QoS routing*, IEEE Communications Magazine, vol. 40, no. 12 (December 2002).
- [32] Likhanov and Mazumdar, *Cell loss asymptotics for buffers fed with a large number of independent stationary sources*, Porc. of IEEE, INFOCOM 98 (1998).
- [33] Neil O'Connell, *Large deviations for departures from a shared buffer*, Journal on Applied Probability (1997), 34: 753–766.
- [34] O. Ozturk, R. Mazumdar, and N. Likhanov, *many sources asymptotics in networks with small buffers*, Queueing Systems (QUESTA) (2003 , to appear.).

- [35] G. Perera, M. Simon, and J. Pechiar, *Effective bandwidth estimation and testing for markov sources*, Eighth IFIP workshop on performance modelling and evaluation of ATM and IP networks (2000).
- [36] P. Rabinovitch, *Statistical estimation of effective bandwidth*, Master's thesis, University of Cambridge., 2000.
- [37] R.Casellas, J.L.Rougier, and D.Kofman, *Packet based load sharing schemes in MPLS networks*, ECUMN'2002. Colmar (April 2002.).
- [38] Thomas G. Robertazzi, *Computer networks and systems - queueing theory and performance evaluation- third edition*, Springer, 2000.
- [39] E. Rosen and A. Viswanathan, *Multiprotocol label switching architecture*, RFC3031, IETF (January 2001).
- [40] Sanjay Shakkottai and R. Srikant, *Many-sources delay asymptotics with applications to priority queues*, Queueing Systems Theory and Applications (QUESTA) **39** (2001), 183–200.
- [41] A. Simonian and J. Guibert, *Large deviations approximations for fluid queues fed by a large number of on/off sources*, IEEE JSAC, 13(7) (August 1995.), 1017–1027.
- [42] P. Srisuresh and P. Joseph, *OSPF-xTE: An experimental extension to OSPF for traffic engineering ;draft-srisuresh-ospf-te-05.txt*, IETF (2003).
- [43] Damon Wischik, *The output of a switch or effective bandwidths for networks*, Queueing Systems **32** (1999).
- [44] Damon Wischik., *Sample path large deviations for queues with many inputs*, Annals of Applied Probability (2000).
- [45] Z.Wang and J.Crowcroft, *Quality of service routing for supporting multimedia applications*, IEEE JSAC (Sept. 1996).