

UT-T00029-SSI
Don.-2012

xx(199624.1)



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Laboratorio de Tecnologías de Información

**Optimización de indicadores de
negocio para servicios de próxima
generación en redes cognitivas**

Tesis que presenta:

Edna Mariel Mil Chontal

Para obtener el grado de:

**Maestro en Ciencias
en Computación**

Director de la Tesis:
Dr. Javier Rubio Loyola

Cd. Victoria, Tamaulipas, México.



Febrero, 2012

UT00029
UT-T00029-SSI
24-08-2012
Jan-2012

ID: 199448-2001

© Derechos reservados por
Edna Mariel Mil Chontal
2012

Esta investigación fue parcialmente financiada mediante el proyecto No. 51623 del Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas.

This research was partially funded by project number 51623 from "Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas"

La tesis presentada por Edna Mariel Mil Chontal fue aprobada por:

Dr. Javier Rubio Loyola, Director

Cd. Victoria, Tamaulipas, México., 24 de Febrero de 2012

Índice General

Índice General	I
Índice de Figuras	v
Índice de Tablas	vii
Índice de Algoritmos	ix
Publicaciones	xi
Resumen	xiii
Abstract	xv
Nomenclatura	xvii
1. Introducción	1
1.1. Antecedentes	1
1.2. Descripción del problema	2
1.3. Hipótesis	4
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos particulares	4
1.5. Metodología	5
2. Estado del Arte	7
2.1. Arquitectura integral para proveer QoS en Internet intradominio	7
2.1.1. Gestión de servicio	10
2.1.1.1. Control de admisión	10
2.1.2. Ingeniería de tráfico	12
2.1.2.1. Dimensionamiento de la Red	14
2.2. Gestión basada en políticas	14
2.3. Refinamiento de Políticas orientado a Calidad de Servicio (QoS)	16
2.3.1. Definición de objetivos de alto nivel	16
2.3.2. Arquitectura de refinamiento de políticas	18
2.4. Técnicas de Optimización	20
2.4.1. Optimización Multiobjetivo	20
2.4.2. Dominancia de Óptimo de Pareto	21
2.4.3. Óptimo de Pareto	22
2.4.4. Frente de Pareto	22

2.4.5.	Técnicas para Manejo de Restricciones Basadas en Conceptos de Optimización Multiobjetivo	22
2.4.6.	Algoritmos evolutivos para optimización multiobjetivo	23
2.4.6.1.	NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm)	24
2.4.6.2.	SPEA (Strength Pareto Evolutionary Algorithm)	25
2.4.6.3.	ParEGO (Pareto Efficient Global Optimization)	25
2.5.	Trabajo Previo	27
2.5.1.	Control de Admisión estático para gestionar QoS en redes Diffserv	28
2.5.2.	Control de Admisión dinámico para gestionar QoS en redes Diffserv	30
2.5.3.	Parámetros de políticas de los BIs en el componente SLS-I	31
2.5.4.	Resumen	34
3.	Enfoque Propuesto	35
3.1.	Arquitectura de gestión para optimización de Indicadores de Negocio	35
3.2.	Metodología de gestión para la optimización de los BIs	39
3.2.1.	Definición de indicadores de negocio relevantes a servicios diferenciados con QoS	40
3.2.2.	Establecer la relación entre los indicadores de negocio con las políticas de gestión en redes cognitivas con QoS	41
3.2.3.	Establecer métricas para evaluar los indicadores de negocio	44
3.2.3.1.	Método para medir el indicador de negocio pérdida por rechazo de invocaciones	44
3.2.3.2.	Método para medir el indicador de negocio degradación del servicio	45
3.2.3.3.	Método para medir el indicador de negocio satisfacción del servicio	48
3.3.	Aplicar métodos de Optimización multiobjetivo para los indicadores de negocio	50
3.3.1.	Uso de modelos surrogados para resolver problemas de optimización multi-objetivo en redes cognitivas de próxima generación	52
3.3.2.	Aplicación de los modelos surrogados para la optimización de los indicadores de negocio en redes cognitivas de próxima generación	54
3.3.3.	Conclusiones	57
4.	Validación Experimental	61
4.1.	Implementación del Enfoque propuesto	62
4.2.	Escenario de validación	63
4.2.1.	Dimensionamiento de Red	66
4.2.2.	Rangos propuestos de las políticas de gestión	69
4.2.3.	Configuración de parámetros del algoritmo ParEGO	70
4.2.4.	Resultados y Análisis	70
4.3.	Conclusiones	74
5.	Conclusiones	75
5.1.	Contribuciones	76
5.2.	Dificultades Técnicas	76

5.3. Conclusiones	77
5.4. Trabajo futuro	78
A. Diagrama de clases	79
A.1. Diagrama de clases del software de gestión de políticas	79
A.2. Diagrama de clases del software de evaluación de los BIs	80
B. Modificaciones en OPNET	83
B.1. Módulos de proceso modificados en OPNET Modeler	83
B.1.1. Proceso mpls_mgr	84
B.1.2. Proceso pl_vm	85
B.1.3. Proceso ip_dispatch	93
B.1.4. Proceso gna_clsvr	94
Bibliografía	97

Índice de Figuras

2.1. Arquitectura general de QoS y la noción del ciclo de aprovisionamiento de recursos.	8
2.2. Arquitectura de Gestión de QoS orientado al negocio	10
2.3. Componentes del control de admisión de invocación.	13
2.4. Arquitectura de políticas de la IETF.	15
2.5. Consideraciones genéricas para definir objetivos de alto nivel.	18
2.6. Arquitectura de refinamiento de políticas orientado a objetivos	19
2.7. Arquitectura de Gestión de QoS orientado al negocio	28
2.8. Acciones de políticas del SLS-S	29
2.9. Búfer de recursos disponibles (RAB)	29
2.10. Acciones de políticas del SLS-I	30
2.11. Relación entre los BIs, objetivos y políticas	31
3.1. Arquitectura de gestión para la optimización de los BIs en redes fijas	36
3.2. Relación entre los BIs, objetivos y políticas	43
3.3. Ejemplo gráfico para medir el BI pérdidas debido al rechazo de invocaciones.	46
3.4. Ejemplo gráfico para medir el BI degradación del servicio.	48
3.5. Topología de la red y parámetros de políticas SLS-I.	55
3.6. Flujo de ejecución de parEGO implementado en el módulo <i>Optimización de Indicadores de Negocio</i>	58
4.1. Integración del software de gestión, la plataforma de prueba, el software que evalúa los indicadores de negocio y el algoritmo que optimiza los indicadores.	62
4.2. Topología de red para el análisis experimental	65
4.3. Frente de Pareto del escenario de prueba 1	70
4.4. Frente de Pareto del escenario de prueba 2	71
4.5. Frente de Pareto del escenario de prueba 3	72
A.1. Diagrama de clases del software de gestión de políticas	80
A.2. Diagrama de clases del software de evaluación de los BIs	82
B.1. Estados del proceso <i>mpls_mgr</i>	84
B.2. Diagrama del proceso <i>pl_vm</i>	85
B.3. Proceso <i>ip_dispatch</i>	94
B.4. Diagrama de estados del proceso <i>gna_clsvr</i>	95

Índice de Tablas

4.1. Perfiles de servicios.	64
4.2. Patrones de tráfico para las aplicaciones por conjunto de servicios.	64
4.3. Resultados de la distancia generacional de cada escenario.	73
4.4. Resultados obtenidos al evaluar los indicadores de negocio mediante asignación de pesos de importancia.	73
4.5. Distancia Euclidiana y Distancia generacional.	74
A.1. Descripción de métodos implementados en el software de gestión de políticas.	81
A.2. Descripción de métodos implementados en la clase BIDegradacion.	81
A.3. Descripción de métodos implementados en la clase BIRechazos.	81
A.4. Descripción de métodos implementados en la clase BISatisfaccion.	82

Índice de Algoritmos

1.	NSGA-II	24
2.	SPEA	25
3.	pseudocódigo: ParEGO	27
4.	pseudocódigo: ParEGO implementado en el módulo Optimización de Indicadores de Negocio.	56

Publicaciones

Optimización de indicadores de negocio para servicios de próxima generación en redes cognitivas

por

Edna Mariel Mil Chontal

Maestro en Ciencias del Laboratorio de Tecnologías de Información
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2012
Dr. Javier Rubio Loyola, Director

Los sistemas de gestión son componentes de software desarrollados para controlar la utilización de los recursos de red y el aprovisionamiento de servicios en las infraestructuras de red. Las redes cognitivas son un nuevo tipo de red en las que se incluyen procesos que ayudan a percibir las condiciones de la red, planear, decidir y actuar en respuesta de esas condiciones, aprender de las consecuencias de acciones previas, tomando en cuenta objetivos globales. La gestión de redes y servicios tradicionalmente está enfocada a satisfacer o controlar aspectos de calidad de servicio. Sin embargo, el último objetivo de los proveedores de redes y servicios es el beneficio económico obtenido a partir del aprovisionamiento de servicios y de la óptima gestión de las redes que los soportan. El aporte fundamental de este trabajo de investigación es proponer una solución de gestión que permita optimizar los indicadores de negocio en redes cognitivas, y que a su vez considere mantener al máximo condiciones óptimas de calidad de servicio a los usuarios. En esta Tesis, se validarán los conceptos desarrollados mediante un software de gestión en el cual se ponga en práctica la solución propuesta. Para evaluar los resultados se tomará en cuenta un escenario de redes cognitivas fijas, para lo cual se hará uso de un ambiente de simulación de redes especializado sobre el cual se integrará la solución desarrollada.

Dynamic optimization of business indicators for next generation in cognitive networks

by

Edna Mariel Mil Chontal

Master of Science from the Information Technology Laboratory
Research Center for Advanced Study from the National Polytechnic Institute, 2012
Dr. Javier Rubio Loyola, Advisor

Management systems are software components developed to control the use of network resources and the provisioning of services in network infrastructures. Cognitive networks are a new type of network that includes processes that help to see the conditions of the network, plan, decide and act in response to these conditions, as well as to learn from the consequences of previous actions, while taking into account the overall objectives. Network management and services are traditionally focused on satisfying or controlling purely quality of service aspects. However, the ultimate goal of network and service providers is the economic revenue from the provisioning of services and the optimal management of networks that support them. The main contribution of this research is to propose a management solution that allows optimize the business indicators in cognitive networks and while delivering the most optimum quality of service to the users. In this Thesis, the developed concepts will be validated through a management software that has been designed to explicitly implement the proposed solution. The results of our proposal will be evaluated with a scenario of fixed cognitive networks, taking into account a specialized simulation networks environment where the developed solution will be integrated.

Nomenclatura

Acrónimos principales

ISP	Proveedores de Servicios de Internet (del Inglés Internet Service Provider)
SLS	Especificaciones del nivel de servicio (del Inglés Service Level Specification)
SLS – S	Especificaciones del nivel de servicio de suscripción (del Inglés SLS-Subscription)
SLS – I	Especificaciones del nivel de servicio de invocación (del Inglés SLS-Invocation)
QoS	Calidad de Servicio (del Inglés Quality of Service)
Intserv	Servicios Integrados (del Inglés Integrated Services)
Differv	Servicios Diferenciados (del Inglés Differentiated Services)
MPLS	Conmutación Multi-Protocolo mediante Etiquetas (del Inglés Multiprotocol Label Switching)
TEQUILA	Traffic Engineering for the Internet at Large Scale
TM	Matriz de tráfico (del Inglés Traffic Matrix)
TT	Traffic trunk
BIs	Indicadores de negocio (del Inglés Business Indicators)
SLAs	Acuerdos de nivel de servicio (del Inglés Service Level Agreements)
SLS	Especificaciones de nivel de Servicio (del Inglés Service Level Specification)
QC	Clase de calidad de servicio (del Inglés QoS Class)
SR_{AVG}	Tasa promedio por tipo de servicio (del Inglés Average rate per Service type)
RAB	Búfer de disponibilidad de recursos (del Inglés Resource Availability Buffer)
TCL	Nivel crítico de objetivo (del Inglés Target Critical Levels)
SR	Tasa de servicio (del Inglés Service Rate)
AC	Control de Admisión (del Inglés Admission Control)
SR_{FS}	Tasa de servicio totalmente satisfecho (del Inglés Service Rate Fully Satisfied)
SR_{AS}	Tasa de servicio casi satisfecho (del Inglés Service Rate Almost Satisfied)
TEQUILA	Traffic Engineering for Quality of Service in the Internet
SrvMgt	Gestión de servicio (del Inglés Service Management)
TE	Ingeniería de tráfico (del Inglés Traffic Engineering)
TF	Traffic Forecast
ND	Dimensionamiento de la red (del Inglés Network Dimensioning)
RAM	Matriz de recursos disponibles (del Inglés Resource Availability Matrix)
IETF	Internet engineering task force
PDPs	Puntos de decisión de políticas (del Inglés Policy Decision Points)
PEPs	Puntos de ejecución de políticas (del Inglés Policy Enforcement Points)
AEs	Algoritmos Evolutivos
CE	Computación Evolutiva
ParEGO	Pareto Efficient Global Optimization
DACE	Design and Analysis of Computer Experiments
MOPs	Problemas de Optimización multiobjetivo (del Inglés Multiobjective optimization Problems)
MOEAs	Algoritmos evolutivos para optimización multiobjetivo (del Inglés Multiobjective Evolutionary Algorithms)
pEAs	Algoritmos Evolutivos Paralelos (del Inglés Parallel Evolutionary Algorithms)

1

Introducción

1.1 Antecedentes

Una red consta de muchas piezas complejas de hardware y software donde interactúan desde enlaces, bridges, routers, host y otros dispositivos, que constituyen los componentes físicos de la red, hasta los diversos protocolos que coordinan y controlan estos dispositivos. El Internet es una red de computadoras mundial, que conecta millones de dispositivos de cómputo a través del mundo. En el lenguaje de Internet, estos dispositivos se llaman host y sistemas terminales. Los sistemas terminales acceden a Internet a través de Proveedores de Servicios de Internet (ISP). Cada ISP es una red de routers y enlaces de comunicación. Los diferentes proveedores proporcionan diferentes tipos de ingreso a la red para los sistemas terminales. Los sistemas terminales, los routers, y otros elementos de Internet, ejecutan protocolos que controlan el envío y la recepción de información en Internet. Internet proporciona un servicio de mejor esfuerzo (best-effort, por sus siglas en Inglés) a todas sus aplicaciones, por lo que no hace ninguna promesa sobre la calidad de servicio que recibirá una aplicación, no proporciona un servicio que haga promesa respecto a qué tiempo se precisará para entregar los datos desde el emisor hasta el receptor y no hay garantías de entrega de

los paquetes. La Calidad de servicio (QoS, por sus siglas en Inglés) en redes IP es un área importante para los proveedores de servicios dado que da lugar a nuevas oportunidades de negocio por ejemplo, diferenciando el tráfico que ingresan los clientes correlacionándolo con su calidad de servicio y el costo.

Existen principalmente dos tipos de tecnologías que proporcionan QoS en Internet. La primera se basa en la reserva y asignación de recursos basándose en flujos de tráfico. Este tipo de tecnología se conoce como Servicios Integrados (Intserv, por sus siglas en Inglés) y ha tenido poca aceptación debido a la escalabilidad en su uso. Alternativamente, un segundo tipo para ofrecer calidad de servicio en Internet se caracteriza por la priorización de determinado tipo de tráfico, en donde los flujos de datos individuales se van agrupando en grandes agregados de tráfico de acuerdo a la *clase de servicio* a la que pertenezcan, y dependiendo de esa clase de servicio recibirán un distinto trato en los diferentes elementos de la red. En esta segunda arquitectura conocida como Servicios Diferenciados (DiffServ, por sus siglas en Inglés) y es en la que está enfocada esta tesis.

Para proveer QoS en Internet es necesario gestionar los servicios que se estarán proporcionando; así como los recursos disponibles en la red. Los sistemas de gestión son componentes de software desarrollados para controlar la utilización de los recursos de red y el aprovisionamiento de servicios en las infraestructuras de redes y servicios. Las redes cognitivas son un tipo de red compuestas de elementos que, a través del aprendizaje y el razonamiento, se adaptan dinámicamente a las diversas condiciones de la red, con el fin de optimizar recursos de extremo a extremo [2]. Tradicionalmente, la gestión de redes y servicios está enfocada a satisfacer o controlar aspectos de calidad de servicio. En la actualidad las redes cognitivas se encuentran en desarrollo, tal es así que los avances significativos en esta temática se orientan al estudio de las diferentes técnicas que habilitan las capacidades de adaptabilidad que demanda el concepto de una red cognitiva y los servicios que éstas soportan.

1.2 Descripción del problema

Internet fue diseñado originalmente para ofrecer el servicio de entrega de datos sin consideraciones de QoS, pero a medida que Internet se ha desarrollado, muchas aplicaciones, particularmente aquellas

aplicaciones de tiempo real, como voz y multimedia, demandan niveles de control sobre el retraso de transmisión y la pérdida de paquetes de información.

Una de las soluciones más significativas para el aprovisionamiento de calidad de servicio en Internet es la propuesta por la solución TEQUILA (Traffic Engineering for the Internet at Large Scale) [14]. En esta solución, el aprovisionamiento de calidad de servicio envuelve la interacción de dos actividades inter-dependientes: Gestión de Servicios e Ingeniería de Tráfico [11]. La Gestión de Servicios es responsable de administrar los acuerdos de nivel de servicio (SLAs) con los clientes y derivar las especificaciones del nivel de servicio (SLSs), mientras que la Ingeniería de Tráfico es responsable de derivar la configuración de la red que satisfaga las demandas de tráfico de los SLSs [11].

En la actualidad es posible controlar el comportamiento de las actividades de gestión de servicios e ingeniería de tráfico mediante políticas de gestión [7]. También es posible derivar políticas que reflejen las decisiones tecno-económicas de un proveedor de servicios diferenciados; en redes soportadas por túneles MPLS durante el aprovisionamiento de los servicios de próxima generación [7]. Sin embargo, las políticas con las que se controlan los dispositivos de red son estáticas, es decir, no cambian en función de nuevas condiciones de utilización de servicios, de las condiciones de la red ni mucho menos del beneficio económico del operador reflejado por los indicadores de negocio. De tal modo, se necesita contar con una solución de gestión que permita optimizar los indicadores de negocio (BIs, por sus siglas en Inglés) de un proveedor de servicios, tomando en cuenta el cumplimiento de las condiciones de calidad de servicio de los clientes mediante una búsqueda de las políticas de gestión adecuadas que consigan que por una parte, se obtengan beneficios económicos adecuados para el operador, y por otra parte, que se mantengan parámetros de calidad de servicio aceptables.

Uno de los problemas centrales a resolver en este trabajo de tesis es la limitación de que los indicadores de negocio del operador y la utilización de recursos son aspectos contradictorios, y por ende, encontrar un punto ideal es prácticamente imposible dado que existen variables tales como la cantidad de usuarios, la invocación de los servicios, el tiempo de invocación de los servicios, la calidad de los servicios, etc, que presentan alta aleatoriedad con lo cual este trabajo afronta la necesidad de encontrar un compromiso entre la Calidad de Servicio y los indicadores de negocio de tal modo que

se optimice la utilización de los recursos y se maximice el beneficio económico en la medida que los usuarios obtengan también la mayor calidad de servicio en todo momento.

1.3 Hipótesis

La Hipótesis de este trabajo de tesis es el siguiente: Es posible optimizar los indicadores de negocio de un proveedor de red y servicios, mediante el uso de algoritmos de optimización multiobjetivo; con la finalidad de mejorar la calidad de servicio en la red y maximizar el beneficio económico de los proveedores de redes y servicios.

1.4 Objetivos

La gestión de red tradicionalmente no toma en cuenta aspectos de negocio. En casos ocasionales las políticas con las que se controlan los dispositivos de red son estáticas, es decir, no cambian en función de nuevas condiciones de utilización de servicios, de las condiciones de la red ni mucho menos del beneficio económico del operador reflejado por los indicadores de negocio.

1.4.1 Objetivo general

El objetivo principal de este trabajo, es proponer una solución de gestión que ayude a obtener el mayor beneficio económico durante el aprovisionamiento de servicios de próxima generación a partir de un análisis y optimización de los indicadores de negocio y de los parámetros de calidad de servicio, soportados por redes cognitivas.

1.4.2 Objetivos particulares

Para lograr el objetivo general, se definieron los siguientes objetivos particulares:

- Contribuir al estado del arte, con una arquitectura de gestión; que habilite la optimización dinámica de indicadores de negocio sobre redes cognitivas.

- Contribuir al estado del arte, con una metodología de gestión; que abarque todos los aspectos necesarios para realizar la optimización de indicadores de negocio, para el aprovisionamiento de servicios de próxima generación sobre redes cognitivas.
- Validar los conceptos y soluciones propuestas en este trabajo de tesis, mediante el desarrollo de un software de gestión; que tome en cuenta las características de redes cognitivas fijas de servicios diferenciados.

1.5 Metodología

La metodología que se ha utilizado para llevar a cabo este trabajo de Tesis se dividió en cinco etapas principales:

- **Revisión bibliográfica de las distintas arquitecturas que proporcionan calidad de servicio.** En esta fase se realizó un estudio del estado del arte, analizando las distintas arquitecturas de gestión para proveer QoS, así como las implicaciones de elevar la gestión de calidad de servicio a aspectos tecno-económicos. En base a esta revisión se pudo escoger la arquitectura de gestión de referencia sobre la cual elevar el concepto de gestión tecno-económica.
- **Elección de los indicadores de negocio (BIs) claves y definición de métricas para evaluar los mismos.** Se realizó una revisión de la literatura y se definieron los BI's aplicables a la invocación de servicios en escenarios de redes fijas. Por medio de estos BIs, un proveedor de red obtendrá información de la situación en que se encuentra la red y podrá tomar acciones correctivas y/o proactivas de acuerdo a sus objetivos globales de optimización. Un aspecto importante a destacar es que los indicadores de negocio y la arquitectura de gestión considerados en estos dos primeros pasos de la metodología tienen una inter-dependencia directa, por lo que su elección ha definido el desarrollo del trabajo de la Tesis. Por ejemplo, las acciones correctivas y/o proactivas consideradas en la arquitectura de gestión tendrán una incidencia determinada en los indicadores de negocio considerados. La definición de métricas por otro lado, define los

parámetros de medida de los indicadores de negocio, sobre los cuales se habrá de optimizar sus valores de una manera cuantitativa.

- **Diseño de la arquitectura de gestión, que habilite la optimización de indicadores de negocio y calidad de servicio en redes cognitivas.** Etapa en la que se integran los elementos necesarios para obtener una arquitectura de gestión autónoma de red, basada en políticas de gestión y en la optimización de los indicadores de negocio.
- **Definición de la metodología de gestión, que hace realidad la optimización tecnológica de gestión con calidad de servicio.** En esta fase se propone una metodología que comprende cuatro etapas fundamentales: 1) Definición de indicadores de negocio relevantes a servicios diferenciados con QoS; 2) establecer la relación entre los indicadores de negocio con las políticas de gestión en redes cognitivas con QoS; 3) establecer métricas para evaluar los indicadores de negocio; 4) aplicar métodos de Optimización multiobjetivo para los indicadores de negocio.

Dentro de este cuarto punto de la metodología se ha hecho un análisis de diversos algoritmos evolutivos con la finalidad de encontrar valores óptimos de las políticas de gestión aplicables a la arquitectura de gestión seleccionada y cuyo funcionamiento en la red con demandas de tráfico específicas y con topologías de red concretas, optimizarán los indicadores de negocio definidos. En base al desempeño mostrado por los algoritmos analizados se seleccionó un algoritmo evolutivo surrogado (ParEGO).

- **Desarrollo de un software de gestión e integración con el software de simulación de red especializado utilizando un escenario de red cognitiva DiffServ sobre MPLS.** El software de gestión fue desarrollado en java y el sistema de dominio Diffserv/MPLS ha sido desarrollado en la plataforma de simulación OPNET. La integración de los dos elementos ha sido empleado para validar la arquitectura y metodología desarrollada en este trabajo de tesis.

2

Estado del Arte

En el presente capítulo se hace un una revisión del estado del arte de las temáticas centrales de esta tesis: la arquitectura de referencia para proveer calidad de servicio en internet en redes DiffServ con MPLS, la arquitectura de gestión basada en políticas, el refinamiento de políticas orientado a calidad de servicio y los conceptos fundamentales de optimización multiobjetivo. Finalmente, se presenta uno de los trabajos más cercanos a la temática de este trabajo de tesis, el cual asocia indicadores de negocio con políticas de gestión para servicios diferenciados.

2.1 Arquitectura integral para proveer QoS en Internet intradominio

El Control de Admisión es un componente clave para la entrega de QoS en redes IP, ya que determina el grado que los recursos de la red son utilizados, y si las características de QoS contratadas son efectivamente entregadas. En [11] los autores posicionan el control de admisión en una arquitectura unificada en donde la ingeniería de tráfico y la gestión de servicio interactúan en diferentes niveles de abstracción y escala de tiempo. Los autores distinguen entre suscripción de servicios e

invocación de los servicios y proporcionan en [14] la arquitectura TEQUILA (Traffic Engineering for Quality of Service in the Internet) la cual es una arquitectura integrada y las técnicas asociadas para automatizar la entrega de QoS en una red IP con capacidad DiffServ sobre MPLS.

La arquitectura general para proveer QoS intradominio, se muestra en la figura 2.1 y se integra de los siguientes subsistemas:

- *Sistema de gestión de servicio (Service management system)*. Es responsable de administrar los acuerdos de nivel de servicio (SLAs) con los clientes y derivar las especificaciones del nivel de servicio (SLSs). También es responsable de gestionar las invocaciones de servicios.
- *Sistema de ingeniería de tráfico (Traffic engineering system)*. Es responsable de derivar la configuración de la red que satisfaga las demandas de tráfico y por ende del cumplimiento de los servicios contratados mediante la derivada configuración de la red.
- *Sistema de monitoreo (Monitoring system)*. Es responsable de proporcionar a los sistemas anteriores, las medidas adecuadas de la red y asegurar que los servicios contratados se entregan realmente con la especificada QoS.

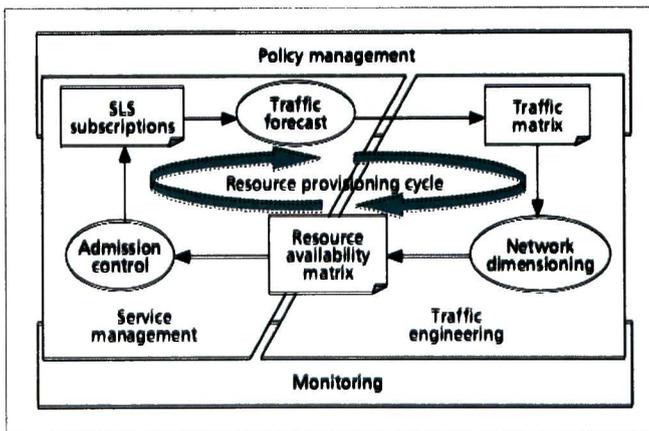


Figura 2.1: Arquitectura general de QoS y la noción del ciclo de aprovisionamiento de recursos.

La arquitectura propuesta anteriormente, establece una clara distinción entre el tratamiento de los clientes (por Especificaciones del nivel de servicio – Service Level Specifications SLS) y el

tratamiento de los recursos (por clases con QoS). La interconexión se definen a través del ciclo de aprovisionamiento de los recursos, (ver figura 2.1). El sistema de gestión de servicios tiene el conocimiento acerca de todos los clientes; pero desconoce de los detalles internos de la red, teniendo sólo un punto de vista de la red como una matriz con capacidades punto a punto.

Los servicios con QoS se ofrecen en términos de los acuerdos de nivel de servicio (SLAs, por su siglas en inglés), los cuales describen todos los aspectos de un contrato de servicio entre un Proveedor de Servicios de Internet (ISP, por sus siglas en Inglés) y sus clientes [11]. Estos acuerdos y en especial las negociaciones que anticiparon el proveedor y sus clientes, se simplifican a través de un conjunto estandarizado llamado especificaciones de nivel de Servicio (SLS, por su siglas en inglés) de [13]. Un SLA contiene los precios, términos, condiciones y los parámetros técnicos del SLS.

El sistema de *gestión de políticas* (ver figura 2.1) se introduce como un medio para guiar el comportamiento de los subsistemas anteriores. Las políticas se definen de una manera declarativa de alto nivel, y se asignan como parámetros y funciones en el sistema de bajo nivel. De esta manera, la inteligencia de los componentes pueden ser dinámicamente ampliados, reducidos o modificados a través de las políticas.

La arquitectura en [14] distingue entre las funciones de servicio y recursos. La capa de recursos emplea gestión de red e ingeniería de tráfico, mientras que la capa de servicio incluye las funciones necesarias para ofrecer y establecer los SLAs, y subsecuentemente manejar la admisión de solicitudes de servicios. A pesar de que distinguen entre la capa de servicios e ingeniería de tráfico, su sistema no actúa de manera aislada. La ingeniería de tráfico proporciona los motivos sobre el cual opera la capa de servicio, por el contrario, la capa de servicio establece los objetivos relacionados con el tráfico para lograr la ingeniería de tráfico. Más específicamente, la capa de servicios proporciona la matriz de tráfico (Traffic Matrix TM, por sus siglas en inglés) a la ingeniería de tráfico, la cual especifica de manera anticipada la demanda de tráfico con QoS entre los bordes de la red. Debido a la relevancia que tiene para la Tesis, a continuación se describe con más detalle el módulo de gestión de servicio de esta arquitectura de calidad de servicio.

2.1.1 Gestión de servicio

La gestión de servicio (SrvMgt, por sus siglas en inglés) maneja los acuerdos de las solicitudes de servicios del cliente, tratando de maximizar el tráfico entrante, en términos del número de contratos y rendimiento, respetando al mismo tiempo el compromiso acordado con el proveedor de servicio (SP, por sus siglas en inglés) de garantizar la QoS.

En la figura 2.2 se muestra el módulo de gestión de servicio de la arquitectura de calidad de servicio TEQUILA, el cual incluye los módulos SLS-Subscription(SLS-S) y SLS-Invocation(SLS-I). El objetivo de la lógica de suscripciones SLS-S es controlar el número y tipo de suscripciones de servicio, con el fin de evitar sobrecarga en la red, mientras que la lógica de invocación SLS-I, su objetivo principal es controlar la cantidad de tráfico que se inyecta en la red.

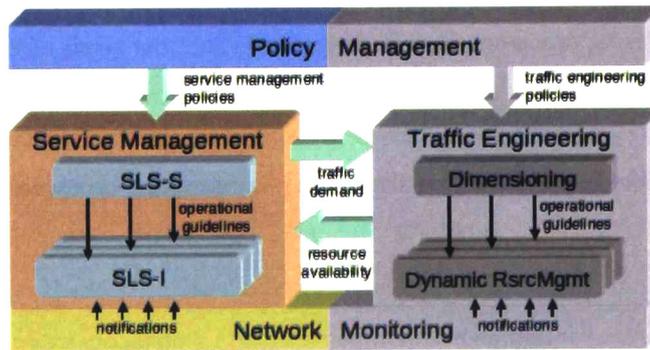


Figura 2.2: Arquitectura de Gestión de QoS orientado al negocio

2.1.1.1. Control de admisión

Control de Admisión se basa en aceptar o rechazar las solicitudes de los servicios en base a la disponibilidad de los recursos de la red. Es una parte integral para la solución de la entrega de QoS en redes IP

En [11] presenta un enfoque de control de admisión de la solución TEQUILA (Traffic Engineering for Quality of Service in the Internet). TEQUILA fue diseñado e implementado como una solución completa para la entrega de QoS en redes IP DiffServ/MPLS. El sistema de control de admisión

utiliza la estimación de los recursos disponibles calculados por la *ingeniería de tráfico*, dando como resultado una matriz de recursos disponibles (RAM, por sus siglas en inglés) por *traffic trunk* (TT, por sus siglas en Inglés). Una traffic trunk representa el tráfico total que pertenece a una clase de servicio con requisitos de QoS semejantes y tiene un determinado alcance topológico (punto a punto de un túnel MPLS). De hecho, las TT son agregados de tráfico con QoS con características de transferencia de la clase asociada con QoS entre los extremos de la red en el dominio del proveedor.

La eficacia del control de admisión depende en gran medida de la carga ofrecida, que a su vez depende de la población suscrita. Por lo tanto, aplicar lógica de suscripción (es decir, regular la población de los suscritos) aumenta el nivel de control y flexibilidad en la solución de control de admisión tradicionales entre la utilización de la red y el deterioro de la QoS. El diseño propuesto en [11] para el control de admisión, se compone en dos enfoques: **1. lógica de suscripción y 2. lógica de invocación.**

1. Lógica de suscripción

El objetivo de la *lógica de suscripción* es determinar si las solicitudes de suscripciones del servicio son aceptadas, con el fin de evitar eventualmente sobrecargar la red, mientras tanto, maximizar el número de suscriptores. Se introduce el parámetro *nivel de satisfacción*, que denota el grado de satisfacción que el proveedor de servicios opta en ofrecer a sus suscriptores.

Dada una solicitud de suscripción, la demanda implícita se deriva y se agrega a la demanda de suscripciones ya establecidas. La demanda total prevista se estima en términos de un valor mínimo y máximo, que corresponde a las tasas de servicios casi y totalmente satisfechos de los servicios suscritos respectivamente.

El nivel de satisfacción determina la parte de las demandas mínimas y máximas de forma segura en la red. Los valores positivos implican que además de la demanda mínima, una parte de la demanda máxima debe ser de forma segura. Los valores negativos implican que sólo una parte de la demanda mínima necesita ser de forma segura.

2. Lógica de invocación

La *lógica de invocación* abarca dos aspectos diferentes:

- Controlar la cantidad y tipo de servicios activos.
- Control del volumen del tráfico inyectado por los servicios activos.

Los objetivos de la lógica de la invocación son:

- Maximizar el número de servicios admitidos y la QoS que disfrutan, maximizando la utilización de la red y evitar la degradación de la QoS causada por la sobrecarga de la red.
- Resolver efectivamente la congestión eventual.

La lógica de invocación de servicios se basa en un modelo retroalimentado, orientado por un evento de monitorización. Mykoniati et al. [11] distinguen entre gestión de admisión dinámica y ejecución de estrategias de admisión (ver figura 2.3). Una estrategia de control de admisión específica la invocación de servicios explícitos, donde la probabilidad de admisión y el porcentaje de la tasa de servicio que se ofrecen a los servicios activos, están en un rango de la tasa de servicio plenamente y casi satisfechos. La gestión de admisión dinámica enfocada a eventos de monitorización, deduce las adecuadas estrategias de control de admisión que serán realizadas por la función de ejecución de estrategias de admisión, esta última interactúa con el cliente para manejar las peticiones de invocación de servicios explícitos y configurar los *routers* frontera, adecuando el tráfico que determinan la tasa de servicio admitido y en consecuencia el grado de satisfacción. Sin embargo, este proceso de gestión de invocación de servicios no toma en cuenta los aspectos económicos de los proveedores de servicios, ni los cambios dinámicos de los indicadores de negocio.

2.1.2 Ingeniería de tráfico

La ingeniería de tráfico (TE, por sus siglas en inglés) se refiere principalmente a la gestión de los recursos de red, con el fin de acomodar el tráfico ofrecido de una manera óptima. La TE dimensiona la red en base a la demanda prevista y del estado actual de la red para entregar QoS.

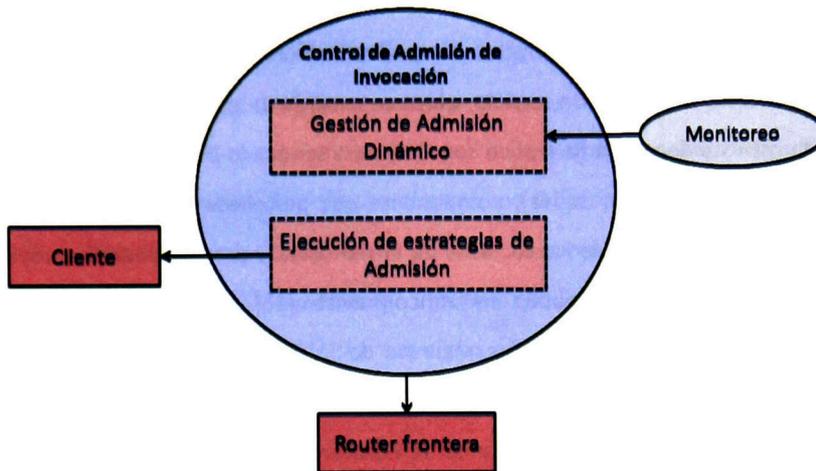


Figura 2.3: Componentes del control de admisión de invocación.

Ingeniería de tráfico es, en general, el proceso de especificar la forma en que el tráfico es tratado en una red determinada. TE tiene tanto el usuario y el sistema orientado a objetivos. Los usuarios esperan que el rendimiento determinado de la red, que a su vez deberían tratar de satisfacer esas expectativas. El rendimiento esperado depende del tipo de tráfico que la red transporta, y se especifica en el contrato de SLS entre el cliente y el proveedor de servicios de Internet (ISP). El operador de red, por el contrario, debe tratar de satisfacer las necesidades de tráfico de los usuarios de forma rentable. Por lo tanto, el objetivo es dar cabida a las solicitudes de tráfico al mayor número posible, de tal manera que se optimice la utilización de los recursos disponibles de la red. Ambos objetivos son difíciles de realizar en un entorno de red multiservicio.

La arquitectura incorpora la capa de servicio dinámico e ingeniería de tráfico (ver figura 2.1), que mediante la utilización del estado actual de la red y la carga de información, intenta optimizar el rendimiento de la red en términos de admisión de servicio y la utilización de recursos, y al mismo tiempo conocer las limitaciones del tráfico con QoS.

2.1.2.1. Dimensionamiento de la Red

En base al contrato SLS de suscripciones, el operador de red utiliza la información declarada en el contrato para el dimensionamiento de la red [13]. La demanda del tráfico se prevee a partir de esos datos históricos, que a su vez son utilizados por la Traffic Forecast (TF, por sus siglas en inglés), módulo que pertenece a la gestión de servicio, quien se encarga de generar la matriz de tráfico (TM) que especifica la prevista demanda de tráfico con QoS para ser usado por la TE. El dimensionamiento de la red (ND, por sus siglas en inglés), componente que pertenece a la ingeniería de tráfico; es responsable de asignar el tráfico de los recursos de la red física y proporcionar el aprovisionamiento de la red con el fin de proveer la demanda del tráfico previsto [13]. ND es responsable de determinar el coste efectivo para la asignación de los recursos de la red física sujeto a las limitaciones de recursos, tendencias de cargas, requisitos de QoS, directivas y restricciones de políticas. Los recursos que se asignan son principalmente limitaciones de enrutamiento con QoS, así como la capacidad del enlace y el espacio del búfer del router, mientras que los medios para la asignación de estos recursos son: capacidad de asignación, mecanismos de ruteo, scheduling y sistemas de gestión del búfer. El componente ND es centralizado para un particular Sistema Autónomo (AS, por sus siglas en inglés), su principal tarea es aceptar como entrada la demanda prevista de la TF, conocer la topología física de la red para calcular la configuración requerida por la elemental TE. Una vez la red ha sido dimensionada, el resultado es una matriz de recursos disponibles (Resource Availability Matrix - RAM, por sus siglas en inglés) que denota los recursos que se cree que estarán disponibles en la red para admitir el tráfico.

2.2 Gestión basada en políticas

La IETF (Internet Engineering Task Force) ha propuesto una arquitectura general de gestión basada en políticas. Una de las motivaciones fundamentales de la gestión basada en políticas es ofrecer flexibilidad y adaptación a los necesarios cambios durante un largo período de tiempo.

Las políticas se definen como un conjunto de reglas para administrar, gestionar y controlar el

acceso a recursos de la red [15]. Estas reglas a su vez se descomponen en eventos, condiciones y acciones. Siempre que se registra un evento de política se evalúa la condición del entorno y si la condición que se especifica en la política se cumple, se ejecuta la acción de dicha política. IETF ha definido una arquitectura que está compuesta por cuatro componentes principales (ver figura 2.4): a) *La herramienta de gestión de políticas* proporciona la interfaz para el administrador de red, para la entrada e implementación de gestión de políticas que será almacenado en un b) *repositorio de políticas* para su futuro uso. Al implementar una política, la herramienta de políticas activa los c) *puntos de decisión de políticas (PDP, por sus siglas en inglés)*, gestor de recursos o servidor de políticas responsable de manejar eventos de políticas, recupera la política desde el repositorio y lo traduce a comandos de configuración en los d) *puntos de ejecución de políticas (PEP, por sus siglas en inglés)*, ubicado dentro de los dispositivos de red. El PDP decide si la condición de una política se cumple y en consecuencia se ejecutan las acciones relacionada con esa política. Este componente también detecta si hay conflictos entre políticas.

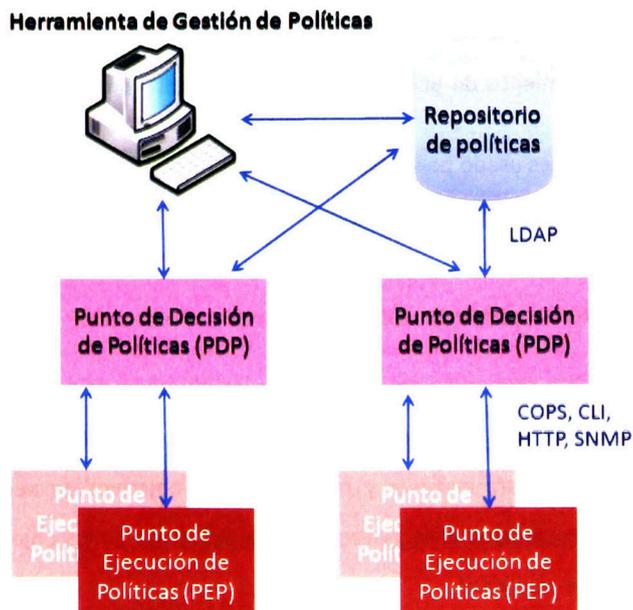


Figura 2.4: Arquitectura de políticas de la IETF.

2.3 Refinamiento de Políticas orientado a Calidad de Servicio (QoS)

Refinamiento de políticas significa derivar políticas de bajo nivel a partir de políticas de alto nivel. La gestión basada en políticas ha sido considerado como un enfoque adecuado para gestionar y controlar sistemas complejos. A pesar de sus potenciales beneficios de flexibilidad y programabilidad, esta solución no es todavía una realidad. Además de los problemas cruciales de conflictos entre políticas, un caso clave detrás de la desconfianza de adoptar esta tecnología, es la necesidad de derivar políticas ejecutables alineadas con las directrices administrativas. Este último normalmente se conoce como el problema de refinamiento de políticas.

En [7] describen un enfoque metodológico para el refinamiento de políticas que está integrado por los siguientes elementos principales:

- Un procedimiento para definir jerarquías de políticas del sistema de políticas.
- Identificación de las directrices de alto nivel.
- Definición de una arquitectura para el refinamiento de políticas que capture los requerimientos para permitir el refinamiento de políticas ejecutables de alto nivel de forma sistemática.

El término *objetivo* en [7] es usado para denotar tanto las directrices administrativas y de negocios. El término *política* se utiliza para ejecutar reglas.

2.3.1 Definición de objetivos de alto nivel

Diferentes métodos se han propuesto para especificar los objetivos de negocio y para proporcionar indicadores para evaluar el desempeño y su relación con ellos. Aunque este es un tema orientado a la aplicación, es imperativo para expresar y representar los objetivos de negocio de una manera accesible. Se describen las consideraciones generales de la definición de los objetivos de negocio, más conocido como objetivos de alto nivel.

En [7] identifican dos actores administrativos: administrador de desarrollo y administrador consultor. El primero define durante el diseño del sistema, los objetivos de alto nivel que el sistema puede manejar y las diferentes maneras de lograrlos. Este grupo también establece la relación entre los objetivos de alto nivel y las políticas que ayudan a lograrlo. Por otro lado, el administrador consultor interpreta los objetivos de alto nivel para definir objetivos "particulares" que reflejen mejor las vistas administrativas para lograr objetivos de alto nivel en el tiempo de operación del sistema.

Un ejemplo de lo anterior se muestra en la figura 2.5. El administrador de desarrollo define durante el diseño del sistema los objetivos de alto nivel "número de suscripciones controladas" para controlar el número de suscripciones del sistema de gestión. Este objetivo es a su vez relacionado con políticas dedicadas al control de las suscripciones de servicios, esto es, políticas de suscripción de servicio. Durante la operación del sistema el administrador consultor interpreta este objetivo de alto nivel y define un objetivo "particular", instanciándolo como "maximizar el número de suscripciones con la máxima confianza para proporcionar QoS de los servicios en tiempo real". Pueden realizarse varias instancias para otros tipos de servicios y para otros objetivos de alto nivel, además, daría lugar a un conjunto de políticas ejecutables que reflejen decisiones administrativas.

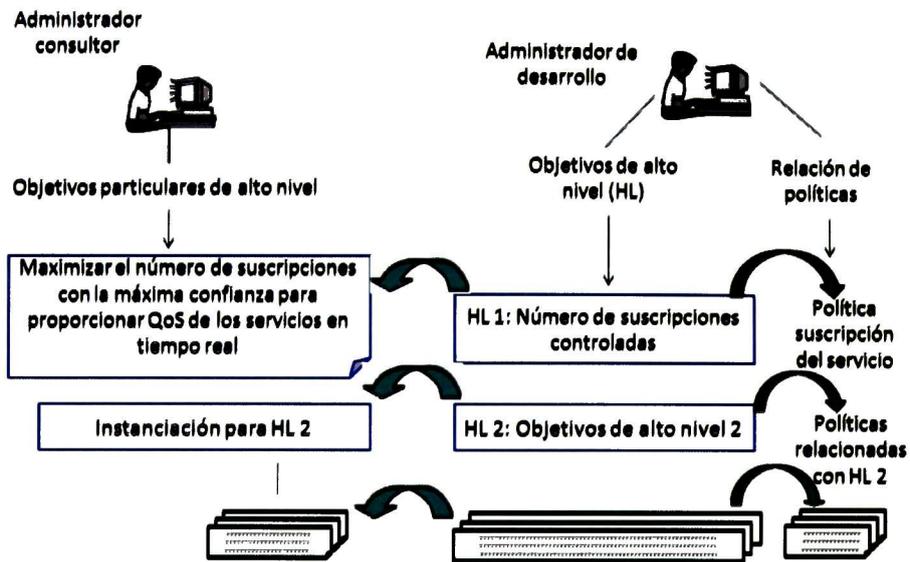


Figura 2.5: Consideraciones genéricas para definir objetivos de alto nivel.

2.3.2 Arquitectura de refinamiento de políticas

La arquitectura de refinamiento de políticas [7] permite refinar políticas ejecutables de más bajo nivel, desde objetivos de alto nivel de una manera sistemática, proporcionar soporte tanto en el sistema de soporte y operación. Se integran dos funciones principales: gestión de objetivos y mecanismos de refinamiento de políticas (ver figura 2.6).

- **Gestión de objetivos** La gestión de objetivos está integrada por la subfunción soporte de refinamiento de objetivos, soporte de selección de objetivos y verificación de objetivos.
- **Soporte de refinamiento de objetivo** - A través del refinamiento de objetivos, el administrador de desarrollo formaliza los requisitos y alternativas con el que los objetivos de alto nivel pueden lograrse. Esta actividad depende de la aplicación y debe ser llevado a cabo durante el diseño del sistema.
- **Soporte de selección de objetivos** A través de la selección de objetivos, el administrador

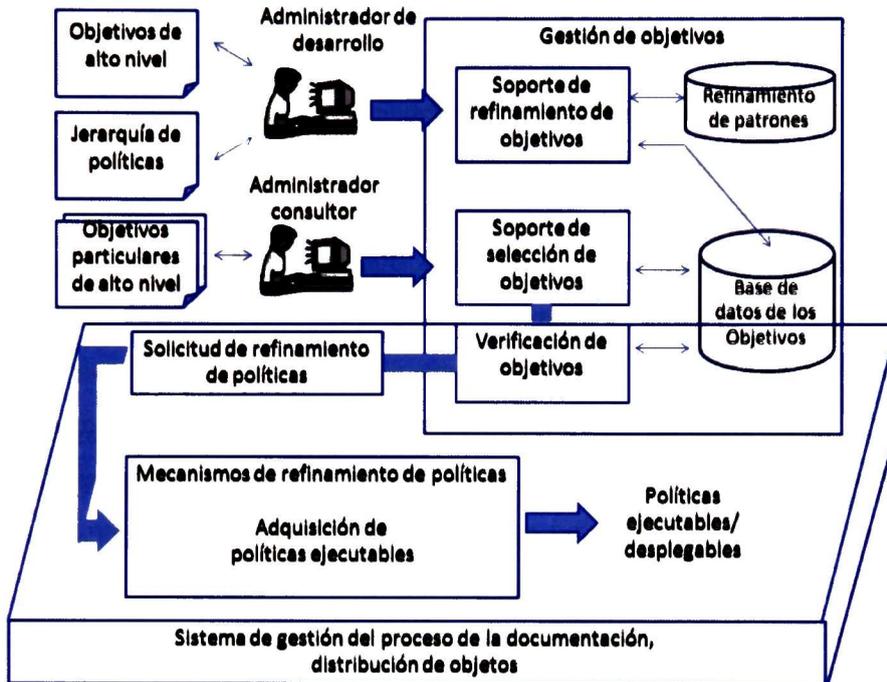


Figura 2.6: Arquitectura de refinamiento de políticas orientado a objetivos

consultor selecciona entre las alternativas, objetivos particulares de alto nivel que reflejan los mejores criterios administrativos en tiempo de operación de servicio.

- Verificación de objetivos - Esta subfunción verifica que la selección de los objetivos sean completas y correctamente lógicas. En otras palabras, verifica que las estrategias seleccionadas ciertamente satisfacen los objetivos particulares de alto nivel. Por ejemplo, el administrador consultor selecciona el objetivo de admisión de suscripción configurado, una verificación simple sería que la selección del objetivo ciertamente incluye ambas mejoras de los objetivos: estimación de la demanda prevista y umbral de admisión configurado.
- Mecanismos de refinamiento de políticas - La función mecanismo de refinamiento de políticas proporciona apoyo para la adquisición de las políticas ejecutables de más bajo nivel que cumplan con los objetivos particulares de alto nivel en tiempo de operación.

2.4 Técnicas de Optimización

2.4.1 Optimización Multiobjetivo

La mayoría de los problemas de optimización del mundo real son naturalmente multiobjetivo, ya que normalmente suelen tener dos o más funciones objetivo que deben satisfacerse al mismo tiempo. Estos problemas son conocidos como (*MOPs: Multiobjective Optimization Problems, problemas de optimización multiobjetivo*). La Optimización Multiobjetivo, también llamada optimización multicriterio, optimización multiperformance u optimización vectorial puede definirse como [3]:

El problema de encontrar un vector de variables de decisión que satisfaga las restricciones y optimice una función vectorial cuyos elementos representan las funciones objetivo. Estas funciones forman una descripción matemática de los criterios de rendimiento que suelen estar en conflicto unas con otras. Por lo tanto, el término "optimizar" significa encontrar una solución en la que todos los objetivos sean satisfechos en un grado aceptable para el tomador de decisiones.

Formalmente, se establece de la siguiente manera:

Encontrar un vector $\bar{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ que satisfaga las m restricciones de desigualdad:

$$g_i(\bar{x}) \geq 0 \quad i = 1, 2, 3, \dots, m \quad (2.1)$$

las p restricciones de igualdad

$$h_i(\bar{x}) = 0 \quad i = 1, 2, 3, \dots, p \quad (2.2)$$

y optimice la función vectorial

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \dots, f_k(\bar{x})]^T \quad (2.3)$$

donde $\bar{x} = [x_1, x_2, \dots, x_n]^T$ es el vector de variables de decisión.

En otras palabras, se desea determinar de entre el conjunto F de valores que satisfacen (2.1) y (2.2) al conjunto $x_1^*, x_2^*, \dots, x_k^*$ que produzca los valores óptimos de todas las funciones objetivo. Las restricciones dadas por (2.1) y (2.2) definen la *región factible* F y cualquier punto \bar{x} en F constituye una *solución factible*. La función vectorial $\bar{f}(\bar{x})$ mapea al conjunto F en el conjunto X , que representa todos los valores posibles de las funciones objetivo. Los k componentes del vector $\bar{f}(\bar{x})$ representa el criterio no conmensurable que debe considerarse. Las restricciones $g_i(\bar{x})$ y $h_i(\bar{x})$ representan las restricciones impuestas a las variables de decisión. El vector \bar{x}^* es reservado para denotar las soluciones óptimas (normalmente habrá más de uno). Esta técnica potencialmente ayudaría a encontrar los valores óptimos de los indicadores de negocio considerando al máximo los requerimientos de calidad de servicio de los usuarios.

2.4.2 Dominancia de Óptimo de Pareto

El concepto Óptimo de Pareto fue formulado por Vilfredo Pareto en el siglo XIX y constituye por sí mismo el origen de la investigación en optimización multiobjetivo. En [3], Coello menciona que un punto $\bar{x}^* \in F$ es un óptimo de Pareto si cada $\bar{x} \in F$ o

$$\bigwedge_{i \in I} (f_i(\bar{x}) \geq f_i(\bar{x}^*)) \quad (2.4)$$

o hay al menos una $i \in I$ tal que

$$f_i(\bar{x}) > f_i(\bar{x}^*) \quad (2.5)$$

Esta definición señala que \bar{x}^* es óptimo de Pareto si no existe un vector factible \bar{x} que disminuiría en algún criterio sin causar un aumento simultáneo de al menos otros criterios. Desafortunadamente, el óptimo de Pareto casi siempre no da una solución única, sino más bien un conjunto de soluciones llamadas soluciones no-inferiores o no-dominadas [3].

2.4.3 Óptimo de Pareto

Un vector solución es considerado *óptimo de Pareto* si no existe otro vector factible que mejore la satisfacción de alguno de los objetivos sin provocar simultáneamente un deterioro en algún otro criterio. Más formalmente, un vector de variables de decisión $\mathbf{X}^* \in \mathcal{F}$ es un óptimo de Pareto si no existe otro vector $\mathbf{X} \in \mathcal{F}$ tal que \mathbf{X} domine a \mathbf{X}^* de acuerdo con la dominancia de Pareto ($\mathbf{X} \prec_P \mathbf{X}^*$) [3].

2.4.4 Frente de Pareto

El conjunto de óptimos de Pareto representado en el espacio de las funciones objetivo constituye lo que se conoce como superficie compromiso o frente de Pareto (FP) [4]:

$$\text{FP} = \{\mathbf{F}(\mathbf{X}^*) : \mathbf{X}^* \in \text{PS}^*\} \quad (2.6)$$

2.4.5 Técnicas para Manejo de Restricciones Basadas en Conceptos de Optimización Multiobjetivo

El interés en [10] es el encontrar el vector \bar{x} que optimice $\bar{f}(\bar{x})$ sujeta a: $g_i(\bar{x}) \leq 0, i = 1, 2, 3, \dots, n;$ $h_j(\bar{x}) = 0, j = 1, 2, 3, \dots, p$ donde $\bar{x} = [x_1, x_2, \dots, x_r]^T$ son las variables de decisión, n es el número de restricciones de desigualdad y p es el número de restricciones de igualdad del problema (en ambos casos las restricciones pueden ser lineales o no lineales). Para utilizar una técnica multiobjetivo para manejar restricciones se debe redefinir el problema mono-objetivo como uno multiobjetivo en el cual se tendrán $m + 1$ objetivos, donde m es el número total de restricciones del problema y el objetivo adicional es la función objetivo original [10]. De esta manera se puede aplicar una técnica multiobjetivo

al nuevo vector $\bar{v} = (f(\bar{x}), f_1(\bar{x}), \dots, f_m(\bar{x}))$, donde $f_1(\bar{x}), \dots, f_m(\bar{x})$ son las restricciones originales del problema.

Existen principalmente tres mecanismos tomados de la optimización evolutiva multiobjetivo que se utilizan para manejar restricciones: (1) Uso de la dominancia de Pareto como un criterio de selección, (2) uso de la jerarquización de Pareto para asignar una aptitud a los individuos y (3) dividir a la población en subpoblaciones que serán evaluadas cada una con respecto a una restricción del problema o a la función objetivo original. Una técnica multiobjetivo está diseñada para encontrar un conjunto de soluciones compromiso que son igualmente buenas entre sí. Por otro lado, en optimización global se busca la solución óptima. Por lo que ésta técnica de optimización multiobjetivo potencialmente podría ayudar a contribuir a encontrar la solución óptima en los indicadores de negocio con sus respectivos objetivos orientados al negocio y proveer QoS.

2.4.6 Algoritmos evolutivos para optimización multiobjetivo

Tal como se mencionó en las secciones anteriores, al optimizar múltiples objetivos estamos interesados en encontrar un conjunto de soluciones compromiso que aproximen de manera representativa el frente de Pareto. Los algoritmos evolutivos para optimización multiobjetivo (MOEAs, por sus siglas en Inglés) han sido exitosamente utilizados para este tipo de problemas. Los MOEAs surgen como una extensión natural de la aplicación de la Computación Evolutiva (CE) o Algoritmos Evolutivos (AEs) a problemas monobjetivo [9].

Los MOEAs tienen como característica principal el contar con una población de posibles soluciones y un proceso de reproducción, el cual permite la combinación de soluciones existentes para generar nuevas soluciones. Los AEs cuentan con una técnica de selección que permite obtener los individuos más aptos para pasar a la siguiente generación permitiendo de esta manera encontrar varios miembros del conjunto de óptimos de Pareto en una sola ejecución del algoritmo [9].

Existen diferentes técnicas para resolver problemas multiobjetivo; en concreto, hablaremos de tres algoritmos evolutivos multiobjetivo considerados en este trabajo de Tesis: **NSGA-II**, **SPEA** y **ParEGO**.

2.4.6.1. NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm)

Propuesto por Deb *et al.* en 2000, se ha convertido en el MOEA más representativo de la literatura y ha sido utilizado ampliamente como punto de referencia para comparativas en nuevos desarrollos. Este algoritmo no trabaja bien al incrementarse la cantidad de objetivos [4].

NSGA-II realiza una clasificación de la población por frentes. Implementa el *Non-Dominated Sorting* (NDS), que es un método de jerarquización basado en la relación de dominancia de Pareto, en otras palabras, los individuos que pertenecen al primer frente son los no dominados; los que pertenecen al segundo frente son los no dominados en ausencia de los del frente anterior, y así sucesivamente. A cada individuo se le asigna un rango equivalente a su nivel de no dominancia. Los mejores individuos son aquellos que tienen rangos menores. Una de las características del **NSGA-II** es la incorporación de un mecanismo explícito para preservación de diversidad, al que los autores nombraron *crowding distance* (CWD). CWD estima la densidad de la región del espacio de objetivos donde se encuentra cada solución. Este operador se utiliza como criterio secundario para discriminar entre soluciones con la misma jerarquía, dando preferencia a aquellas ubicadas en regiones menos densas. El algoritmo 1 describe el pseudocódigo de **NSGA-II**.

Algoritmo 1 NSGA-II

- 1: $gen \leftarrow 0$
 - 2: Generar aleatoriamente población inicial de N padres: P_{gen}
 - 3: Jerarquizar P_{gen} usando *Non-Dominated Sorting*
 - 4: **while** $gen < g_{max}$ (repetir durante g_{max} generaciones) **do**
 - 5: Torneo binario basado en la jerarquía de las soluciones. Si dos soluciones tienen la misma jerarquía, utilizar CWD como criterio de preferencia.
 - 6: Aplicar cruza y mutación para generar la población de N hijos: Q_{gen}
 - 7: Combinar poblaciones de padres e hijos: $R_{gen} \leftarrow P_{gen} \cup Q_{gen}$ ($|R_{gen}| = 2N$)
 - 8: Jerarquizar R_{gen} usando *Non-Dominated Sorting*
 - 9: Seleccionar N individuos de R_{gen} (con respecto a la jerarquía) para formar la población de padres para la siguiente generación, P_{gen+1} . Si existen más soluciones con la misma jerarquía que la capacidad disponible en P_{gen+1} , seleccionar sólo $N - |P_{gen+1}|$ individuos utilizando CWD como criterio de discriminación.
 - 10: $gen \leftarrow gen + 1$
-

2.4.6.2. SPEA (Strength Pareto Evolutionary Algorithm)

El algoritmo Evolutivo de Fuerza de Pareto fue introducido por el Zitzler y Thiele [16]. Es una técnica que integra diferentes algoritmos evolutivos multiobjetivo, buscando combinar lo mejor de cada uno de ellos. Este algoritmo utiliza dos poblaciones, incorporando el concepto de elitismo a través del almacenamiento de las soluciones no dominadas en una población externa, la cual participa del procedimiento de selección. Además, el cálculo del valor de aptitud se realiza utilizando un procedimiento basado en la asignación de un valor de fuerza (strength) a todos los elementos de la población externa. Este procedimiento induce la formación de nichos a partir del concepto de dominancia de Pareto, llamado niching por strength. Puesto que el conjunto de soluciones en la población externa puede ser grande y ésta interviene en el proceso evolutivo, también se utiliza un procedimiento de clustering para reducir el número de soluciones en dicho conjunto. El algoritmo 2 describe el algoritmo básico de SPEA.

Algoritmo 2 SPEA

- 1: Generar aleatoriamente población inicial P .
 - 2: Evaluar cada solución.
 - 3: Crear el conjunto no dominado externo vacío E
 - 4: **while** el criterio de parada no sea alcanzado **do**
 - 5: Copiar el conjunto de soluciones no dominados de P a E
 - 6: Eliminar las soluciones de E cubiertas por cualquier otro miembro de E
 - 7: **if** El número de soluciones en el almacenamiento externo excede la máxima capacidad de E
 then
 - 8: Decidir los elementos a eliminar por medio de clustering
 - 9: Evaluar los individuos de P y E
 - 10: Aplicar la selección del torneo binario para seleccionar los individuos de $P + E$
 - 11: Aplicar los operadores de mutación y de cruce
-

2.4.6.3. ParEGO (Pareto Efficient Global Optimization)

J. Knowles [8] propuso parEGO como una nueva alternativa a la optimización global eficiente (Efficient Global Optimization - EGO, por sus siglas en inglés). El algoritmo EGO hace uso de un proceso llamado Kriging para modelar el paisaje de búsqueda con soluciones conocidas. Más específicamente, explota una versión del modelo DACE (Design and Analysis of Computer Experiments - DACE, por

sus siglas en inglés), basado en procesos Gaussianos para modelar las funciones objetivo [6]. El modelo DACE es una buena opción para construir el tipo de aproximación que se necesita por un algoritmo de búsqueda restringida a un bajo número de evaluaciones de la función.

El algoritmo EGO inicia generando un número de soluciones mediante *hipercubo latino* a lo que llaman diseño experimental space-filling, y encontrar la máxima probabilidad modelo DACE (haciendo uso de algún algoritmo de optimización). Para generar una nueva solución a evaluar, EGO busca la solución que maximiza lo que Jones y otros llaman la *mejora esperada*. La parte de la curva del error estándar en el modelo que se encuentra por debajo del mejor valor actual. Lo que significa que EGO pondera tanto el valor predicho de soluciones, como el error en la predicción, con el fin de encontrar el que tiene el mayor potencial para mejorar el costo mínimo. Sin embargo, no sólo elige la solución que el modelo predice que reducirá al mínimo el valor. Por el contrario, de forma automática balancea la explotación y exploración.

Una vez que la superficie DACE inicial es ajustada y se hizo alguna transformación (si fue necesario), se procede iterativamente. Primero se maximiza la mejora esperada utilizando un algoritmo de branch and bound. Si la mejora esperada es menor que el 1 % del mejor valor actual de la función (en la escala sin transformaciones), entonces termina. De lo contrario, se muestrea la función en donde la mejora esperada fue maximizada, obteniendo una nueva solución que es evaluada (con la verdadera función costosa). El modelo DACE es actualizado con esta nueva información, y la próxima solución se elige utilizando este modelo actualizado. El algoritmo 3 describe el flujo de ejecución de ParEGO.

El algoritmo ParEGO es una extensión del algoritmo EGO para problemas de optimización multi-objetivo. En el enfoque propuesto se convierten los k diferentes valores de una solución en un único valor por medio de una ponderación paramétrica escalar del vector (parameterized scalarizing weight vector). Se elige un diferente vector ponderado en cada iteración de la búsqueda. Una aproximación a la totalidad del frente de Pareto es construida gradualmente.

Algoritmo 3 pseudocódigo: ParEGO

```

DEFINE ParEGO( $f, d, k, s$ )
1:  $xpop[ ] \leftarrow LATINHYPERCUBE(d)$ 
2: for each  $i$  in 1 to  $11d - 1$  do
3:    $zpop[i] \leftarrow EVALUATE(xpop[i], f)$ 
4: while not finished do
5:    $\lambda \leftarrow NEWLAMBDA(k, s)$ 
6:    $model \leftarrow DACE(xpop[i], zpop[ ], \lambda)$ 
7:    $xnew \leftarrow EVOLALG(model, xpop[ ])$ 
8:    $xpop[ ] \leftarrow xpop[ ] \cup \{xnew\}$ 
9:    $znew[ ] \leftarrow EVALUATE(xnew, f)$ 
10:   $zpop[ ] \leftarrow zpop[ ] \cup \{znew\}$ 
END

```

2.5 Trabajo Previo

Uno de los trabajos que se relaciona con nuestro proyecto de tesis es el propuesto en [12], el cual describe un enfoque de control relevante a los indicadores de negocio y evalúan su relación con objetivos y políticas de gestión de servicio. En este trabajo el administrador asigna pesos de importancia a cada BI, los cuales son usados para derivar los parámetros apropiados para las políticas.

La arquitectura que ellos proponen está compuesto por tres distintos bloques: gestión de servicio, ingeniería de tráfico y gestión de políticas. El primero es responsable del acuerdo con los clientes o par de dominios con requerimientos de QoS en términos de SLSs. La ingeniería de tráfico cumple con los SLSs contratados para derivar la configuración de la red. La gestión de políticas guía el comportamiento de los dos bloques anteriores con un conjunto de políticas que reflejen las metas y objetivos.

El enfoque orientado al negocio presentado en [12] se centra en la funcionalidad proporcionado por el bloque de gestión de servicio, en este bloque se encuentran los componentes de la arquitectura TEQUILA: SLS-S y SLS-I, (ver figura 2.7), módulos que realizan control de admisión estático y dinámico respectivamente. El primero tiene una funcionalidad centralizada fuera de línea y realiza el control de admisión de las solicitudes de suscripciones, en base a la disponibilidad de recursos (resource availability) proporcionado por el sistema TE, mientras que el SLS-I se distribuye a través

de los routers de entrada y realiza la invocación dinámica de los SLSs ya suscritos en función del tiempo de ejecución de la red, siguiendo las directrices operativas previsto por SLS-S.

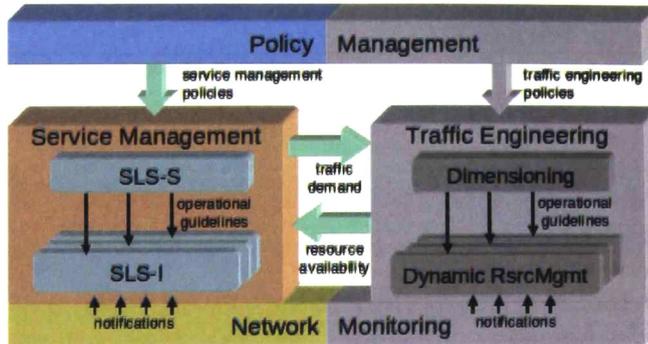


Figura 2.7: Arquitectura de Gestión de QoS orientado al negocio

2.5.1 Control de Admisión estático para gestionar QoS en redes Diffserv

La lógica de suscripción basada en políticas lleva a cabo el control de admisión estático de la cantidad y tipo de suscripciones de servicios, con el fin de evitar sobrecargar la red y al mismo tiempo maximizar el tráfico de los suscriptores. Las políticas que logran estos objetivos se muestran en la figura 2.8. Las acciones P1.1 y P1.2, usan la noción de satisfacción del servicio y niveles de calidad, para establecer los parámetros relevantes por clases con QoS (QC, por sus siglas en inglés). Los parámetros de satisfacción $\epsilon[0, 0.5]$ definen los factores de multiplexación que se utilizan para derivar las tasas en las que un servicio se considera casi satisfecho (AS, por sus siglas en Inglés) y totalmente satisfecho (FS, por sus siglas en Inglés). El servicio se considera *plenamente satisfecho* cuando los usuarios gozan de la QoS de su tasa contratada y *casi satisfechos* cuando gozan de la QoS a tasas inferiores de su tasa contratada, pero por encima de lo que se considera aceptable para el tipo de servicio. Ellos operan a una tasa inicial promedio (SR_{AVG}) por tipo de servicio que es sugerido por el proveedor. Un incremento en el factor AS produce tasas bajas, definido por la expresión ($SR_{AVG} * (1 - Fctr_{AS})$). Un incremento en el factor FS da como resultado tasas más altas, definido por la expresión ($SR_{AVG} * (1 + Fctr_{AS})$).

Políticas SLS-S		
ID	Acción de Políticas	Descripción
P1.1	setAlmstSatisf(QC,Fctr _{AS}) setFullSatisf(QC, Fctr _{FS})	Establece los factores casi y plenamente satisfecho por QC
P1.2	setQlLvl(QC,OQL)	Establece el nivel de calidad por QC
P1.3	setUpprLmt(TT, SU)	Establece los límites superiores del RAB por TT
P1.4	setACStrg(QC,TD)	Establece el AC, decisión de aceptar/rechazar por QC

Figura 2.8: Acciones de políticas del SLS-S

La disponibilidad de los recursos se mide mediante un búfer de disponibilidad de recursos (RAB, por sus siglas en inglés), que mantiene la demanda agregada de la SLSs suscritos por TT. Como se ilustra en la figura 2.9. Los parámetros del RAB son los siguientes:

- R_{min}^a Representa el ancho de banda disponible para una TT garantizado por la red en todo momento.
- R_{min}^w Representa el ancho de banda disponible para una TT garantizado por la red en tiempos de congestión. Este ancho de banda reservado puede ser utilizado por otras TTs, pero cuando hay congestión en la red, la TE forza a todas las TT a tomar el límite de ancho de banda que les corresponde R_{min}^w .
- R_{max} Representa el máximo ancho de banda disponible por las TTs, pero sin garantía de confianza.

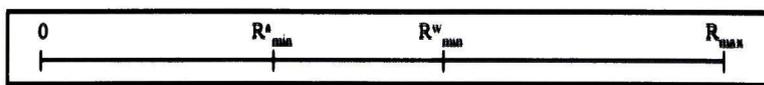


Figura 2.9: Búfer de recursos disponibles (RAB)

2.5.2 Control de Admisión dinámico para gestionar QoS en redes

Diffserv

En contraste a la naturaleza estática del componente de suscripción, la lógica de la invocación de servicios se basa en eventos de tiempo de ejecución para regular el tráfico que entra en la red. Las políticas utilizadas en [12] se muestran en la figura 2.10 y se aplican en el módulo SLS-I. Se lleva a cabo control de admisión dinámica del número de servicios activos, así como del volumen de tráfico admitido. La política de acción P2.1 de la figura 2.10, define un umbral de nivel crítico objetivo (TCL, por sus siglas en Inglés), donde $TCL \in [R_{min}^a, R_{max}]$. Cuanto más cerca esté de R_{min}^a se toman más pronto las acciones proactivas, mientras que valores cercanos a R_{max} resultan retrasos de acciones proactivas.

La operación en tiempo de ejecución del módulo SLS-I es lanzado por el cruce del TCL, el cual activa las políticas de acción P2.2 y P2.3 para ajustar la tasa de servicio (SR, por sus siglas en Inglés) y el umbral control de admisión dinámico (AC_{th} , por sus siglas en Inglés) de un TT. $AC_{th} \in [R_{min}^a, R_{max}]$ controla las invocaciones de los servicios ya suscritos. Cuanto más bajo es, menos posibilidades hay que un SLS entrante tenga éxito. Una nueva solicitud de servicio será aceptada sólo si el uso actual de la TT junto con la tasa promedio de los servicios no exceda el AC_{th} . La $SR \in [AS, FS]$ es el nivel de satisfacción asignado a un TT.

Políticas SLS-I		
ID	Acción de Políticas	Descripción
P2.1	setTCL(TT, TCL)	Establece el umbral de nivel crítico objetivo del RAB por TT
P2.2	setSR(TT, SR)	Establece la tasa de servicio de una TT
P2.3	setAC _{th} (TT, AC)	Establece los límites del control de admisión del RAB por TT

Figura 2.10: Acciones de políticas del SLS-I

2.5.3 Parámetros de políticas de los BIs en el componente SLS-I

En la figura 2.11 [12] muestra la relación que hay entre los indicadores de negocio con sus respectivos objetivos asociados, asignando pesos de importancia para cada BI: *pérdida por rechazo de invocaciones*, *satisfacción del servicio* y *degradación del servicio*. Para establecer valores a los parámetros de las políticas: TCL , AC_{th} y SR , se definieron unas funciones de asignación que consideran pesos de importancia para cada BI.

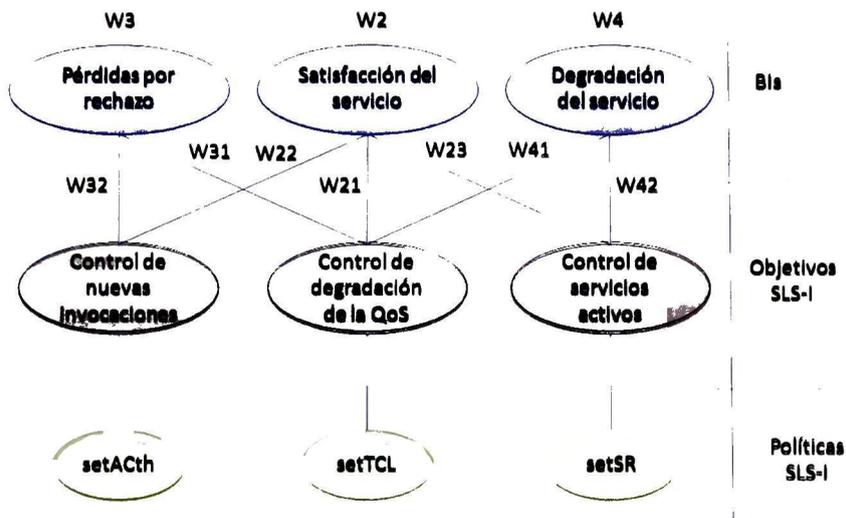


Figura 2.11: Relación entre los BIs, objetivos y políticas

Para establecer valores para el TCL , el cual logra el objetivo control de la degradación de la QoS, objetivo influenciado por los tres BIs, figura 2.11, donde se consideran tres casos: a) Si el TCL está cercano a R_{max} resultan retardos de acciones para prevenir la degradación de la QoS. Esto puede permitir que los servicios activos disfruten de más altas tasas de su tasa promedio por más tiempo; y mantener una alta probabilidad de aceptar nuevas invocaciones. Como tal, la satisfacción del servicio es máxima, y la pérdida debido al rechazo de invocaciones se reduce al mínimo. Estas condiciones, sin embargo, pueden llegar a causar congestión en la red y la degradación del servicio resulta en un potencial de fuertes sanciones. b) Un TCL cercano a R_{min}^a puede tener el efecto contrario en la satisfacción y la pérdida por rechazo de invocaciones, ya que las acciones proactivas

se aplican antes de tiempo. La degradación del servicio también es alta en esta opción, ya que es probable que los servicios reciban menos que su tasa promedio contratada, c) R_{min}^w es considerado como el óptimo valor del TCL para minimizar la degradación del servicio. Esto puede resultar en niveles medios de satisfacción del servicio y pérdidas de rechazos de invocaciones. Formalmente se muestran las funciones de asignación que toman en cuenta los pesos de los tres BIs y derivan los valores apropiados para el TCL.

$$TCL_1 = R_{min}^a + 2W_{21}(R_{min}^w - R_{min}^a), \text{ cuando } W_{21} \leq 0.5 \quad (2.7)$$

$$TCL_1 = (2R_{min}^w - R_{max}) + 2W_{21}(R_{max} - R_{min}^w), \text{ cuando } W_{21} > 0.5 \quad (2.8)$$

$$TCL_2 = R_{min}^a + 2W_{31}(R_{min}^w - R_{min}^a), \text{ cuando } W_{31} \leq 0.5 \quad (2.9)$$

$$TCL_2 = (2R_{min}^w - R_{max}) + 2W_{31}(R_{max} - R_{min}^w), \text{ cuando } W_{31} > 0.5 \quad (2.10)$$

$$TCL_3 = R_{min}^a + W_{41}(R_{min}^w - R_{min}^a), \text{ cuando } W_{41} > 0.5 \quad (2.11)$$

$$TCL_3 = R_{max} + W_{41}(R_{max} - R_{min}^w), \text{ cuando } W_{41} \leq 0.5 \quad (2.12)$$

La política en relación con el objetivo de control de nuevas invocaciones fija el umbral de AC, el cual es influenciado por los indicadores de negocio: satisfacción del servicio y el indicador pérdida por rechazo de invocaciones. Los valores cercanos a R_{max} implica una baja probabilidad de rechazos de invocaciones resultando mínimas pérdidas y una mayor satisfacción. Valores bajos para el AC resulta mayores pérdidas y menor satisfacción, debido a la mayor probabilidad de rechazo de invocaciones, R_{min}^a representa el caso extremo. Las siguientes funciones determinan el AC de acuerdo a los pesos que se les han asignado a los BIs.

$$AC_{th1} = R_{min}^a + 2W_{22}(R_{min}^w - R_{min}^a), \text{ cuando } W_{22} \leq 0.5 \quad (2.13)$$

$$AC_{th1} = (2R_{min}^w - R_{max}) + 2W_{22}(R_{max} - R_{min}^w), \text{ en otro caso} \quad (2.14)$$

$$AC_{th2} = R_{min}^a + 2W_{32}(R_{min}^w - R_{min}^a), \text{ cuando } W_{32} \leq 0.5 \quad (2.15)$$

$$AC_{th2} = (2R_{min}^w - R_{max}) + 2W_{32}(R_{max} - R_{min}^w), \text{ cuando } W_{32} > 0.5 \quad (2.16)$$

Por último, la política que implica los ajustes para el SR de los servicios activos, tienen un impacto en la calidad del servicio percibida por los usuarios y las sanciones aplicables como resultado de la degradación del rendimiento. Valores cercanos a SR_{FS} implican altos niveles de satisfacción y prevención de penalización a los usuarios. La tasa AS puede llevar a la inversa, debido a incumplidas tarifas contratadas. El impacto de los dos BIs que involucran el parámetro SR son cuantificados por las funciones de asignación que se establecieron. Formalmente:

$$SR_1 = SR_{AS} + W_{23}(SR_{FS} - SR_{AS}) \quad (2.17)$$

$$SR_2 = SR_{AS} + W_{42}(SR_{FS} - SR_{AS}) \quad (2.18)$$

Al especificar pesos de importancia a los BIs y utilizar las funciones de asignación descrito, una red puede ser configurada de acuerdo con los objetivos de negocio. Sin embargo, las políticas generadas a partir de dichos pesos producirán resultados económicos de acuerdo a las condiciones de la red, los patrones de invocación de los usuarios, y otros factores imponderantes. El resultado de la ejecución de las políticas en dichas condiciones se mantendrá hasta que se haga una nueva asignación en el valor de las políticas, es decir, se mantendrá invariante en el tiempo. Es por esto que en el presente trabajo de tesis decidimos trabajar en el módulo SLS-I con la finalidad de proveer de una funcionalidad de optimización en los tres indicadores de negocio que se han mencionado anteriormente, encontrando los mejores valores para las políticas de gestión que optimicen los BIs de acuerdo a las condiciones de la red y los recursos disponibles, manteniendo la mejor calidad de servicio posible en todo momento.

2.5.4 Resumen

En este capítulo se presentaron los avances más significativos de los temas relacionados con esta tesis. Se ha presentado una arquitectura de referencia para proveer calidad de servicio en Internet en redes DiffServ con MPLS, la arquitectura de gestión basada en políticas del IETF, el concepto de refinamiento de políticas orientado a calidad de servicio (QoS) y los conceptos fundamentales de optimización multiobjetivo. Finalmente, se ha presentado uno de los trabajos más cercanos a la temática de este trabajo de tesis [12], el cual asocia indicadores de negocio con políticas de gestión para servicios diferenciados.

En [12] definen un conjunto de funciones que toman en cuenta el impacto de los BIs en la gestión de servicio de políticas. El administrador asigna pesos de importancia a cada BI, y con base en estos indicadores de negocio se obtienen las políticas apropiadas. Sin embargo, este proceso de gestión orientado a negocio no toma en cuenta los aspectos económicos de los proveedores de servicios de una manera dinámica, de tal manera que los indicadores de negocio y las acciones correctivas que se ejecutan para cumplir y controlar estos indicadores de negocio, se definen estáticamente y no se ajustan en función de las necesidades de los usuarios, de los mismos objetivos de negocio, ni de otras consideraciones de vital importancia para su optimización.

Hasta el momento no hay reportado ningún trabajo que permita optimizar los indicadores de negocio en redes cognitivas fijas. Es por ello que el enfoque que se plantea en nuestro trabajo de tesis contribuye en gran manera al estado del arte.

3

Enfoque Propuesto

En este Capítulo se describe la solución de este trabajo de Tesis, la cual consiste en una Arquitectura y una Metodología de gestión para la optimización de Indicadores de Negocio.

3.1 Arquitectura de gestión para optimización de Indicadores de Negocio

En esta Sección se describe la arquitectura que integra los elementos necesarios para optimizar los indicadores de negocio y proveer calidad de servicio en Internet; en un dominio de servicios diferenciados en redes fijas. Esta arquitectura ayuda a mantener un equilibrio en los Indicadores de Negocio (BIs) definidos y la calidad de servicio.

La figura 3.1 muestra la arquitectura que se ha propuesto en este trabajo de Tesis, la cual ha sido elaborada tomando en cuenta la arquitectura de la solución TEQUILA (Traffic Engineering Quality of Service for the Internet at Large Scale) [14].

Al igual que la arquitectura TEQUILA, la arquitectura propuesta cuenta con los sub-sistemas, todos ellos con las mismas funcionalidades que la original definida en la solución TEQUILA: **Sistema**

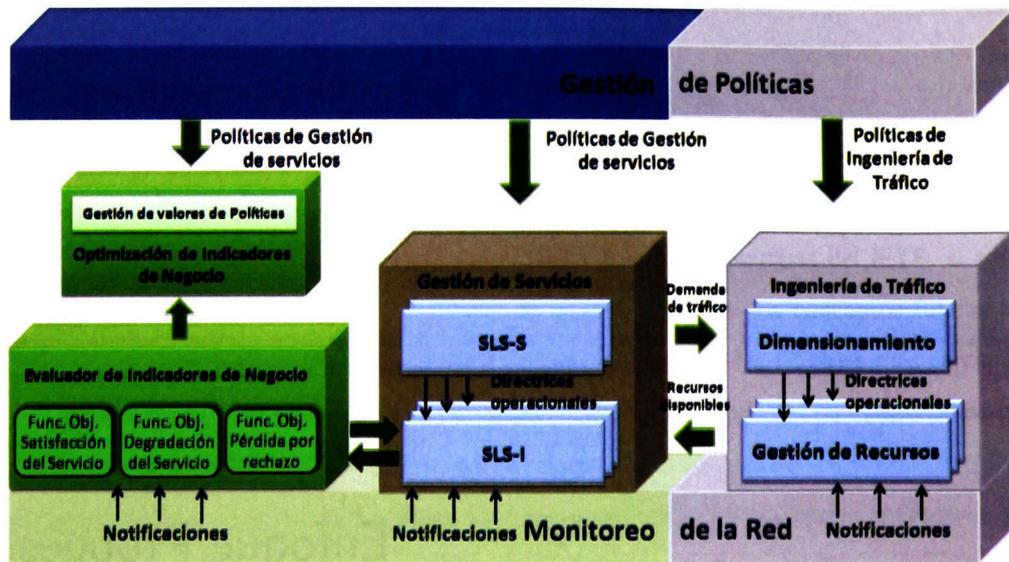


Figura 3.1: Arquitectura de gestión para la optimización de los BIs en redes fijas

de gestión de servicio, Sistema de ingeniería de tráfico, Sistema de monitoreo, Sistema de gestión de políticas.

- **Sistema de gestión de servicio.** Es responsable de administrar los acuerdos de nivel de servicio (SLAs) con los clientes y derivar las especificaciones del nivel de servicio (SLSs). El módulo de gestión de servicio incluye los módulos SLS-Subscription (SLS-S) y SLS-Invocation (SLS-I). El objetivo de la lógica de suscripciones SLS-S es controlar el número y tipo de suscripciones de servicio, con el fin de evitar sobrecarga en la red, mientras que la lógica de invocación SLS-I, su objetivo principal es controlar la cantidad de tráfico que se inyecta en la red.
- **Sistema de ingeniería de tráfico.** Es responsable de derivar la configuración de la red que satisfaga las demandas de tráfico y por ende del cumplimiento de los servicios contratados mediante la derivada configuración de la red. La ingeniería de tráfico se refiere principalmente a la gestión de los recursos de red, con el fin de acomodar el tráfico ofrecido de una manera

óptima. La TE dimensiona la red en base a la demanda prevista y del estado actual de la red para entregar QoS.

- **Sistema de monitoreo.** Es responsable de proporcionar a los sistemas anteriores, las medidas adecuadas de la red y asegurar que los servicios contratados se entregan realmente con la especificada QoS.
- **Sistema de gestión de políticas.** Se introduce como un medio para guiar el comportamiento de los subsistemas anteriores. Las políticas se definen de una manera declarativa de alto nivel, y se asignan como parámetros y funciones en el sistema de bajo nivel. De esta manera, la inteligencia de los componentes puede ser dinámicamente ampliada, reducida o modificada a través de las políticas. En este sentido, las políticas son los medios para habilitar los procesos cognitivos en las redes de próxima generación.

Ahora bien, dada la naturaleza del problema planteado en este proyecto de Tesis ha sido necesario agregar a esta arquitectura dos módulos para introducir y realizar un manejo adecuado de los Indicadores de Negocio en esta arquitectura de gestión: **Evaluador de Indicadores de Negocio y Optimización de Indicadores de Negocio.**

- **Evaluador de Indicadores de Negocio.** Este proceso tiene en cuenta los Indicadores de Negocio definidos por el proveedor de redes y servicios y los evalúa en función del contexto en el que se encuentre el funcionamiento de la red y el aprovisionamiento del servicio y su calidad. Los Indicadores de Negocio sirven a los proveedores de redes y servicios para saber y/o definir los objetivos de alto nivel y saber si éstos se están cumpliendo a lo largo del aprovisionamiento de un servicio, con la finalidad de decidir las estrategias a implementar en determinado momento.

Los indicadores de negocio son la representación formal de los objetivos de negocio del proveedor en lenguaje de alto nivel, de tal modo que éstos deben ser medibles. Los BIs se establecen con base en las metas y objetivos para el éxito del negocio. En la arquitectura de la figura 3.1 se muestra el módulo *Evaluador de Indicadores de Negocio (BIs)* con los tres BIs que se

han considerado en este trabajo de Tesis: *Satisfacción del servicio, Degradación del servicio y Pérdidas por Rechazo*.

Debido a la naturaleza de estos tres indicadores, el proceso *Evaluador de Indicadores de Negocio* está ligado al módulo SLS-I de la arquitectura propuesta, debido a que estos tres indicadores se ven afectados por el comportamiento de la red de acceso, y es precisamente el módulo SLS-I el módulo que controla dicha funcionalidad en nuestra arquitectura de gestión. Es decir, los Indicadores de Negocio que hemos considerado en este trabajo de Tesis se ven directamente afectados por las funcionalidades básicas que controla el módulo SLS-I de la arquitectura de gestión:

- Controlar la cantidad y tipo de servicios activos.
- Control del volumen del tráfico inyectado por los servicios activos.
- Controlar el número de servicios admitidos y la Calidad de Servicio (QoS) que disfrutan.
- Maximizar la utilización de la red.
- Evitar la degradación de la Calidad de Servicio de los usuarios debido a la sobrecarga de la red.
- Resolver las hipotéticas congestiones.

Cabe destacar que el módulo SLS-I se distribuye a través de los routers de ingreso de la red DiffServ del proveedor de redes y servicios, y su comportamiento es definido por políticas de gestión. Cuando es necesario hacer una modificación al comportamiento de algún módulo SLS-I de una manera dinámica, este carga con la nueva configuración en el router de acuerdo a las políticas que se hayan definido para dicha modificación.

En resumen, el módulo *Evaluador de Indicadores de Negocio* evalúa el estado actual de las características que controlan los módulos SLS-I en cada uno de los routers frontera de la red DiffServ de cada Traffic Trunk MPLS y evalúa su incidencia en los Indicadores de Negocio (BIs) con la finalidad de obtener un valor cuantitativo en dichos BIs y poder tener valores cuantitativos comparables de dichos Indicadores de Negocio.

- **Optimización de Indicadores de Negocio.** Tiene la tarea de encontrar las políticas de configuración de los módulos SLS-I ideales que logre un “compromiso” en los valores cuantitativos de los Indicadores de Negocio obtenidos en la evaluación. Para ello, realiza un procedimiento basado en los siguientes aspectos:

Este módulo incluye un sub-sistema administrador de valores de políticas para los módulos SLS-I de los routers frontera, con los cuales se realizan las evaluaciones de los indicadores de negocio (realizadas por el módulo *Evaluador de Indicadores de Negocio*). Después de cada evaluación el módulo de Optimización capta los resultados de ejecución del comportamiento de la red con los valores de las políticas propuestas en dicha evaluación. Con los resultados cuantitativos de cada evaluación, el módulo de Optimización tiene los elementos necesarios para definir “las mejores soluciones” y construir un “frente de Pareto” durante un conjunto de evaluaciones. Una ejecución (o corrida) está constituida por un conjunto de evaluaciones. El módulo de *Optimización de Indicadores de Negocio* produce un conjunto de soluciones óptimas (frente de Pareto) por ejecución, a partir del cual, un proveedor de servicios puede elegir el área de operación que mejor le satisfaga tomando en cuenta sus intereses. Las soluciones encontradas en el frente de Pareto por el módulo de *Optimización de Indicadores de Negocio* tienen las siguientes características:

1. Consideran la topología de la red empleada y su dimensionamiento.
2. Consideran la demanda de tráfico utilizada en la ejecución.
3. Considera el tipo de servicios y el patrón de invocación de cada uno de los servicios que se han disfrutado en la red.
4. Consideran todos los valores de las políticas del módulo SLS-I.

3.2 Metodología de gestión para la optimización de los BIs

Una vez definidos los elementos de la arquitectura de gestión y sus funcionalidades básicas, en esta Sección se describe la metodología propuesta para optimizar los Indicadores de Negocio haciendo

uso de los elementos de dicha arquitectura. La metodología básicamente está compuesta por cuatro etapas fundamentales: 1) Definición de Indicadores de Negocio relevantes a servicios diferenciados con QoS; 2) Establecer la relación entre los indicadores de negocio con las políticas de gestión en redes cognitivas con QoS; 3) Establecer métricas para evaluar los Indicadores de Negocio; y 4) Aplicar métodos de Optimización multiobjetivo para los Indicadores de Negocio. En las siguientes subsecciones se explica la metodología de una manera más detallada.

3.2.1 Definición de indicadores de negocio relevantes a servicios diferenciados con QoS

Un Indicador de Negocio (BI, por sus siglas en Inglés) informa a un proveedor de redes y servicios cómo se está comportando la red, respecto al negocio y su situación relativa con referencia a los objetivos estratégicos y metas establecidas. El primer paso de la metodología consiste en definir los Indicadores de Negocio sobre los componentes que la arquitectura contempla para tal fin, el *Evaluador de Indicadores de Negocio y Optimización de Indicadores de Negocio*. Como se ha comentado anteriormente, en este trabajo de tesis se ha enfocado el trabajo hacia Indicadores de Negocio que tienen incidencia o que se ven afectados por las invocaciones de servicios. Los BIs [12] que se han considerado para nuestro trabajo de tesis fueron tomados de la literatura, éstos se consideran los más adecuados para la invocación de los servicios y son los siguientes:

1. **Pérdida por rechazo de invocaciones.** El propósito de este Indicador de Negocio es cuantificar y/o medir las pérdidas debido a las peticiones de servicio que no son aceptadas en la red.
2. **Degradación del servicio.** Este Indicador de Negocio mide la degradación del rendimiento del servicio debido a la congestión en la red, si los usuarios tienden a recibir una calidad inferior que su tasa promedio contratada.
3. **Satisfacción del servicio.** La finalidad de este Indicador de Negocio es medir la satisfacción del servicio que se le proporciona al usuario durante todo instante de tiempo.

3.2.2 Establecer la relación entre los indicadores de negocio con las políticas de gestión en redes cognitivas con QoS

Una vez definidos los Indicadores de Negocio con los que el operador de servicios pretende trabajar, la metodología propuesta considera particularmente relevante establecer una relación entre los Indicadores de Negocio con las políticas de gestión de calidad de servicio (QoS) de la red cognitiva. A continuación se describe la relación de los Indicadores de Negocio con las políticas de gestión de la solución propuesta.

Las políticas de gestión reflejan las metas y objetivos de alto nivel del proveedor de redes y servicios, políticas que se definen a bajo nivel y tienen una interpretación sin ambigüedades; con la capacidad de ser ejecutadas de manera autónoma, de tal modo que cumplan con los objetivos establecidos.

Dado que nuestros objetivos de negocio inciden directamente sobre las invocaciones de servicios, en esta Sección se describe el impacto de las políticas de gestión (también conocido como controles de gestión) aplicables a la invocación de servicios SLS-I con relación a los dichos Indicadores de Negocio. A través de un proceso de refinamiento de políticas orientadas al negocio en redes fijas, los Indicadores de Negocio han sido transformados a objetivos de red de alto nivel y eventualmente, estos objetivos son convertidos a políticas de bajo nivel [7]. Es decir, los Indicadores de Negocio tomados de la literatura se han asociado con objetivos de nivel de red/servicio y a partir de estos se derivan las políticas relevantes.

Como hemos mencionado anteriormente, el uso de políticas ayuda a lograr la cognitividad en la red y obtener un comportamiento de autonomía, puesto que proporcionan los mecanismos de adaptabilidad que necesitan dichas redes de próxima generación, de esta manera, la inteligencia de los componentes de la red pueden ser ampliados de manera dinámica, reducido o modificado a través de las políticas.

Las políticas de control pertenecientes a la invocación de servicios SLS-I se describen a continuación:

1. **TCL (Objetivo de nivel crítico).** Es un umbral que define el nivel en el que se considera la posibilidad de sobrepasar el tráfico de red crítica de una Troncal de Tráfico (Traffic Trunk TT, por sus siglas en Inglés). Una TT representa el tráfico total que pertenece a una clase de servicio con requisitos de QoS semejantes y tiene un determinado alcance topológico (punto a punto de un túnel MPLS). De hecho, las TT son agregados de tráfico con QoS con características de transferencia de la clase asociada con QoS entre los extremos de la red en el dominio del proveedor. Cuando se cruza el umbral de la política *TCL* se toman acciones proactivas que ayudarán a prevenir la degradación de la QoS. Las acciones proactivas pueden ser, la disminución de la tasa de servicio de clientes activos (ver política SR abajo) y/o rechazar usuarios hasta que el tráfico esté por debajo del nivel definido por un umbral determinado (ver política *AC_{th}* abajo).
2. **SR (Tasa de servicio).** Esta política se activa cuando se sobrepasa el umbral *TCL* en una TT. Esta política ayuda a regular las tasas de tráfico que gozan los usuarios activos asignados a una TT. Cuando se sobrepasa el *TCL* de una TT, esta política asigna un nuevo valor de Service Rate a los usuarios que están haciendo uso de esa TT. Un usuario activo es aquél que está haciendo uso de los recursos de la red a través de la TT.
3. ***AC_{th}* (Control de Admisión).** Esta política se activa cuando se sobrepasa el umbral *TCL* en una TT. Cuando se sobrepasa el *TCL* de una TT, esta política comienza a negar el acceso a los usuarios "nuevos" que hacen invocaciones a la red. El umbral que define esta política tiene la finalidad de indicar a partir de qué punto comenzará a aceptar nuevamente a los usuarios que estén invocando los servicios a los cuales fueron suscritos, dado que se espera que el rechazo de usuarios provoque una eventual reducción en la inyección de tráfico en la red. Esta política controla las invocaciones de los usuarios ya suscritos a los servicios que se ofrecen por el proveedor.

Cabe destacar que los routers frontera de la red del operador implementan, por cada Troncal de Tráfico (TT), estos tres tipos de políticas (*TCL*, *SR* y *AC_{th}*).

Estas políticas de invocación controlan el número de usuarios y tipo de servicios; y consecuentemente, el volúmen del tráfico que entra en la red. De esta manera, se compensa entre el número de invocaciones admitidos y se previene la degradación de la QoS debido a la sobrecarga de la red.

La figura 3.2 muestra la relación resultante que hay entre los Indicadores de Negocio y las políticas de control. Se observa una asociación entre los Indicadores de Negocio (BIs), las políticas SLS-I descritas anteriormente, y unos objetivos intermedios de alto nivel a los cuales éstas políticas ayudan a lograr.

La política *TCL* (centro de figura 3.2) ayuda a lograr el objetivo de *control de degradación de la QoS*, el cual está relacionado con los tres Indicadores de Negocio. Por otra parte, la política *AC_{th}* (izquierda de figura 3.2) ayuda a lograr el objetivo *control de nuevas invocaciones*, está influenciado por los indicadores de negocio *satisfacción del servicio* y *pérdida por rechazo de invocaciones*. Por último, la política *SR* (derecha de figura 3.2) implica el ajuste de la tasa de servicio de los usuarios activos y ayuda a lograr el objetivo de controlar los servicios activos. Los valores asignados a esta política tendrán un impacto en los dos indicadores de negocio: *satisfacción del servicio* y *degradación del servicio*.

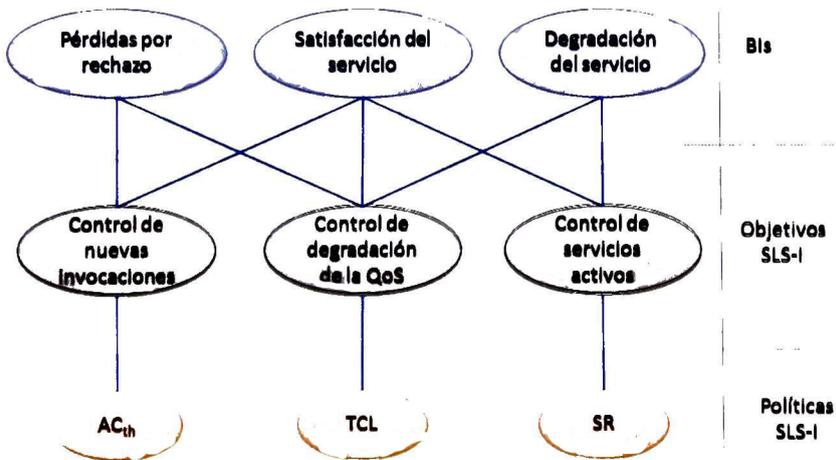


Figura 3.2: Relación entre los BIs, objetivos y políticas

Como hemos descrito anteriormente, los Indicadores de Negocio (BIs) y las políticas SLS-I están

inter-relacionadas. Los indicadores de negocio de los proveedores de redes y servicios evolucionan, de acuerdo a la cantidad de usuarios, a la calidad que éstos reciben durante la utilización de sus servicios, a los tipos de servicios que invocan los usuarios, al patrón de utilización de servicios, etc. De tal modo que, afrontamos la necesidad de definir un conjunto de métricas para evaluar los indicadores de negocio que nos permita abordar con garantías el reto de su optimización. Este reto se describe en la siguiente parte de nuestra metodología.

3.2.3 Establecer métricas para evaluar los indicadores de negocio

Una de las principales características de un Indicador de Negocio es que debe ser cuantificable. Los módulos el *Evaluador de Indicadores de Negocio* y *Optimización de Indicadores de Negocio* de la arquitectura de gestión necesitan llevar a cabo la cuantificación de los indicadores de negocio considerados en nuestro trabajo de Tesis. En esta Sección se describen los métodos de cuantificación para los tres indicadores de negocio (BIs) en relación a las políticas de invocación de servicios de calidad de servicio (QoS) SLS-I de nuestra arquitectura.

3.2.3.1. Método para medir el indicador de negocio pérdida por rechazo de invocaciones

El indicador de negocio *pérdida por rechazo de invocaciones*, representa el número total de usuarios para cada TT que se les ha negado el servicio al que ellos fueron suscritos. Este indicador de negocio está relacionado con dos de las políticas de gestión, AC_{th} y el TCL , tal como se puede observar en la figura 3.2.

Es deseable para un operador de servicios el minimizar el número de usuarios rechazados que invocan los servicios que se les ha proporcionado, por ende aceptar el mayor número de nuevas invocaciones. Con ello, repercutiría positivamente el valor de negocio de la empresa. Sin embargo es necesario cuantificar este valor numérico. Para medir este indicador de negocio se ha realizado lo siguiente:

1. Cuando el umbral de la política TCL es activado, se ejecutan las acciones correctivas para evitar congestión en la red. Uno de los efectos por defecto de esta política (TCL) es el rechazar

usuarios nuevos en la TT en donde se ha registrado el cruce del umbral del TCL , es decir, la activación de la política AC_{th} .

2. Una vez cruzado el umbral TCL y activado el control de acceso AC_{th} se procede a contabilizar el número de usuarios que fueron rechazados por la TT, obteniendo un total de los mismos durante el tiempo que hubo cruce del TCL .

En la figura 3.3 se muestra el método de medida de este Indicador de Negocio. Para ejemplificar esta métrica, se ha colocado el valor de la política TCL en un valor de 16.8Mbps, cuando este límite sea alcanzado, la acción de la política TCL se activa, comenzando a rechazar usuarios nuevos. En ese momento se comienza a cuantificar el número de usuarios rechazados que ha intentado ingresar a la red y que se les ha negado acceso a la misma en esa Troncal de Tráfico. De la misma manera se observa que se ha colocado el valor de la política AC_{th} en un valor de 17.3Mbps. La política AC_{th} se activa cuando el umbral TCL ha sido sobrepasado y consecuentemente se ha cruzado el AC_{th} . En la figura se puede apreciar que como resultado de la ejecución de ambas políticas, después de un tiempo determinado, el tráfico comienza a disminuir y cuando esto ocurre, el tráfico cruza el umbral AC_{th} de forma descendente, por lo que se comienza a aceptar nuevas invocaciones de servicio. En ese momento el conteo de usuarios rechazados se detiene hasta que el efecto mencionado anteriormente se vuelva a presentar en la Troncal de Tráfico. Cabe destacar que este conteo se realiza para cada Troncal de Tráfico y que el módulo *Evaluador de Indicadores de Negocio* realiza los pasos anteriormente descritos y el tratamiento de esta información de una manera adecuada y proporciona los resultados de su evaluación al modulo *Optimización de Indicadores de Negocio*.

3.2.3.2. Método para medir el indicador de negocio degradación del servicio

La degradación del servicio ocurre cuando los usuarios activos no disfrutan las tasas contratadas, y es provocada mayormente cuando los usuarios activos sobrepasan las capacidades de la red. En nuestra solución el método propuesto para medir la degradación del servicio considera los elementos de la arquitectura de calidad de servicio SLS-I. Concretamente, este aspecto está controlado por el umbral de la política TCL en cada troncal TT. Cuando este umbral es sobrepasado se toman

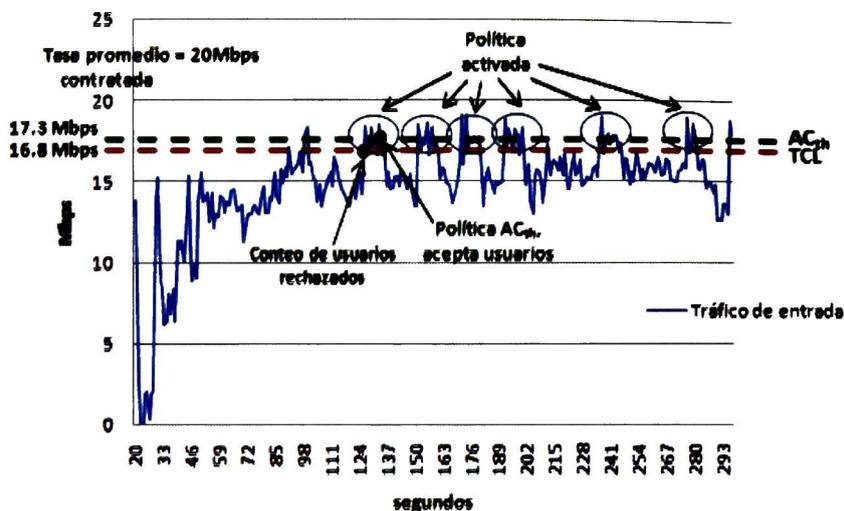


Figura 3.3: Ejemplo gráfico para medir el BI pérdidas debido al rechazo de invocaciones.

acciones correctivas para evitar el congestionamiento en la red. Estas acciones están enfocadas a disminuir la tasa de tráfico a todos los usuarios que pertenecen a una TT, sin embargo, esto puede ocasionar que los usuarios activos reciban menos que su tasa contratada y la degradación del servicio incrementa. Este indicador se relaciona con dos de las políticas de gestión, *TCL* y el *SR* (ver figura 3.2).

Los pasos para medir éste indicador de negocio se describen a continuación:

1. Cuando el umbral de la política *TCL* se encuentra activado en una Troncal de Tráfico TT, se mide el tráfico de entrada a la red. Posteriormente se calcula el porcentaje que corresponde ese tráfico de entrada respecto a la tasa promedio contratada.
2. Se calcula el total de usuarios activos para la TT en cuestión y se pondera el valor calculado con el número de SLAs a los cuales da servicio dicha TT.
3. Finalmente se obtiene el porcentaje que corresponde a este indicador de negocio; mediante el valor obtenido en el paso 1 entre el valor ponderado del paso 2.

4. Se establece una ventana de tiempo. Si esta ventana de tiempo propuesta no ha sido excedida, no se considera degradación en el servicio.

La ecuación (3.1) muestra los pasos descritos anteriormente para medir el Indicador de Negocio *degradación del servicio*.

$$\%DegradacionServicio_{TT} = \frac{(TR_{inTT} \div tasa\ contratada_{TT})}{(usuarios\ activos_{TT} \div SLAs_{TT})} \quad (3.1)$$

Un aspecto importante a destacar es el método para medir el Indicador de Negocio *degradación del servicio* se realiza para cada Troncal de Tráfico (TT) por el *Evaluador de Indicadores de Negocio*, el cual realiza los pasos indicados de una manera adecuada proporcionando los resultados de su evaluación al modulo *Optimización de Indicadores de Negocio*.

La figura 3.4 muestra un ejemplo para ejemplificar la medición del *indicador de degradación de servicio*. En este caso el primer evento de cruce de la política *TCL* se detecta alrededor del segundo 124, con 14 usuarios activos en ese instante de tiempo. Dado esa condición, se calcula el porcentaje correspondiente a la degradación del servicio mediante la ecuación (3.1). Donde la variable TR_{in} corresponde al tráfico de entrada que se está inyectando en la red. *SLAs* representa el número de contratos que se realizó y para el cual esta dimensionada la red y por consiguiente la Troncal de Tráfico (TT). La *tasa promedio contratada* es la tasa de servicio adquirida a partir de los *SLAs* y los *usuarios activos* son los usuarios que están haciendo uso de los recursos de la red en la TT.

El porcentaje del tráfico entrante (TR_{in}) durante ese instante de tiempo es de 18.4Mbps. Considerando una tasa promedio contratada de 24Mbps para 7 *SLAs*, se obtiene lo siguiente:

$$DegradacionServicio = \frac{18.4Mbps \div 24Mbps}{(14 \div 7)} = 0.38333$$

El porcentaje obtenido para este indicador *degradación del servicio* equivale al 38.33% en el segundo 124. Si el periodo de tiempo que ha transcurrido supera a la ventana de tiempo propuesta, se considere degradación en el servicio, de lo contrario no hay degradación. En la figura 3.4 en el segundo 95 aproximadamente ciertamente se ha superado la política de *TCL* pero no se ha superado

la ventana de tiempo, por lo que no se considera que haya habido degradación y por ende no se ha procedido a medir el indicador de negocio.

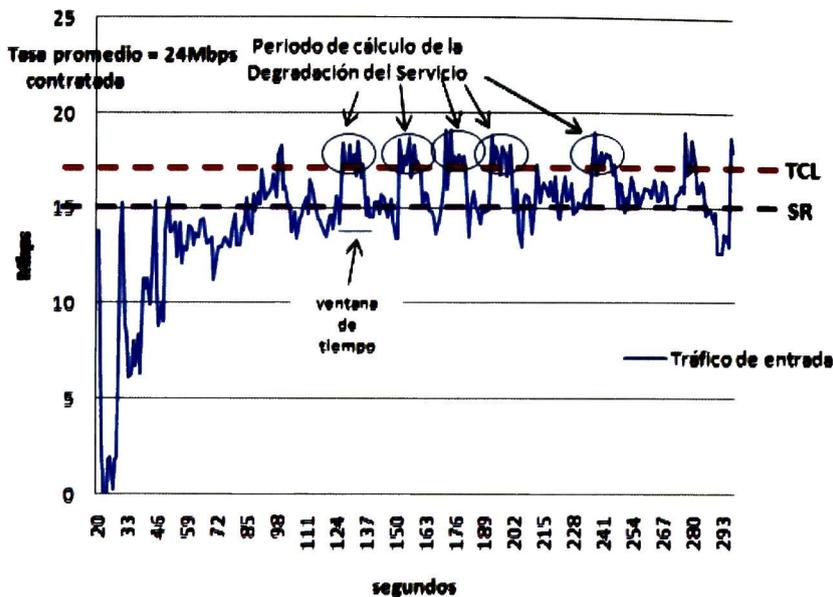


Figura 3.4: Ejemplo gráfico para medir el BI degradación del servicio.

3.2.3.3. Método para medir el indicador de negocio satisfacción del servicio

Este indicador de negocio representa cuantitativamente la satisfacción del servicio que se le proporciona a los usuarios activos. Un aspecto fundamental para medir este indicador ha sido determinar si realmente el proveedor de redes y servicios está proporcionando los recursos disponibles a los clientes, de acuerdo a su tasa de servicio promedio contratada. En este sentido, existe un aspecto muy importante a considerar cuando los clientes de una Troncal de Tráfico están inyectando tráfico inferior a su tasa contratada, dado que, esto puede ser, o bien, por voluntad propia, o bien por falta de recursos de la red del proveedor. Para resolver esta problemática y medir satisfactoriamente este indicador, en este trabajo se han planteado los siguientes aspectos para medir este indicador:

Los pasos que se han planteado para medir este indicador de negocio son los siguientes:

- La satisfacción del servicio se mide en todo tiempo, desde que los usuarios comienzan a enviar tráfico en la red por el Troncal de Tráfico (TT) hasta que finalizan.
- Para verificar si la satisfacción del servicio está siendo proporcionada a los usuarios activos, se mide la capacidad de los enlaces que están siendo compartidos por la TT en cuestión.
- La capacidad que pueden soportar los enlaces fijos para el recorrido del tráfico a su destino es limitado, por lo que al medir el enlace que comparten las TT obtenemos información relevante para conocer si realmente el proveedor está cumpliendo con la satisfacción del servicio a los clientes, por lo que es necesario contemplar dos situaciones; la primera es verificar si hay recursos disponibles en la red para el envío de paquetes y la segunda es comprobar si los usuarios realmente no están haciendo uso de los recursos.

Consideramos un 85% de la capacidad de un enlace compartido por una TT para indicar que aún existen recursos disponibles en la red, de lo contrario, el enlace está siendo utilizado a su nivel máximo, por lo que la satisfacción del servicio comienza a degradarse. El porcentaje estimado se considera un valor aceptable para el envío del tráfico, debido a que estos tipos de enlaces en redes fijas pueden transmitir tráfico sin congestión.

- Si el enlace fijo que comparten las TT esta siendo totalmente demandado al 85%, indica que si están enviando poco tráfico es por que la red no les permite enviar más debido a que los usuarios están haciendo uso de los recursos, por lo que se procede a medir el Indicador de Negocio de satisfacción del servicio de la manera siguiente:
 1. Cuando el umbral de la política *TCL* se encuentra activado en una Troncal de Tráfico TT, se mide el tráfico de entrada a la red. Posteriormente se calcula el porcentaje que corresponde ese tráfico de entrada respecto a la tasa promedio contratada.
 2. Se calcula el total de usuarios activos para la TT en cuestión y se pondera el valor calculado con el número de SLAs a los cuales da servicio dicha TT.
 3. Finalmente se obtiene el porcentaje que corresponde a este indicador de negocio; mediante el valor obtenido en el paso 1 entre el valor ponderado del paso 2.

4. Se establece una ventana de tiempo. Si esta ventana de tiempo propuesta no ha sido excedida, no se considera degradación en el servicio.

- Si el enlace que está siendo compartido por las TT, no es utilizado al 85% entonces la satisfacción del servicio equivale a 1, lo que significa que los usuarios están totalmente satisfechos.

La ecuación (3.2) calcula el porcentaje que corresponde a la satisfacción del servicio, la cual es calculada cuando los enlaces compartidos por el TT en cuestión son utilizados más del 85%.

$$\%Satisfaccion\ del\ Servicio_{TT} = \frac{(TR_{inTT} \div tasa\ contratada_{TT})}{(usuarios\ activos_{TT} \div SLA_{sTT})} \quad (3.2)$$

Un aspecto importante a destacar es el método para medir el Indicador de Negocio satisfacción del servicio se realiza para cada Troncal de Tráfico (TT) por el *Evaluador de Indicadores de Negocio*, el cual realiza los pasos indicados de una manera adecuada proporcionando los resultados de su evaluación al modulo *Optimización de Indicadores de Negocio*.

3.3 Aplicar métodos de Optimización multiobjetivo para los indicadores de negocio

En la Sección 3.2.2 se proporcionó una descripción de la relación que existe entre los Indicadores de Negocio con las políticas de gestión. El enfoque alcanzado a día de hoy en el estado del arte [12] es el dar prioridades a los Indicadores de Negocio para derivar analíticamente las políticas SLS-I en función de pesos asignados a cada Indicador de Negocio sin considerar ningún valor cuantitativo. Al especificar pesos de importancia a los indicadores de negocio y utilizar las funciones de asignación (ver Sección 2.5.3), una red puede ser configurada de acuerdo con los objetivos de negocio. Sin embargo, las políticas derivadas a partir de dichos pesos son estáticas, y no consideran elementos dinámicos como la demanda de servicios, el tipo de servicios, la simultaneidad de los servicios invocados, el patrón de invocación de los servicios, etc. En otras palabras, las políticas no cambian hasta que se

haga una nueva asignación en el valor de los pesos de los indicadores. El estar asignando pesos a cada uno de los indicadores de negocio para encontrar las políticas adecuadas que impacten idealmente en el modelo del negocio del proveedor, es una tarea muy complicada. El espacio de búsqueda de todas las posibles soluciones candidatas al problema, es demasiado amplio debido a que, además de los aspectos dinámicos que hemos mencionado arriba, los valores de las políticas TCL , AC_{th} y SR de cada uno de los módulos SLS-I de los routers frontera de la red tienen valores ilimitados, y no todas las soluciones posibles poseen las características que un proveedor de red desearía.

A diferencia del estado del arte, en donde no se han cuantificado los Indicadores de Negocio, en la Sección anterior se han descrito los aspectos fundamentales para llevar a la práctica la cuantificación de los objetivos de negocio considerados en nuestro trabajo de Tesis. Ahora bien, cabe recordar que estos objetivos de negocio están afectados por los valores que se asignen a las políticas de gestión de los módulos SLS-I, y por el comportamiento de la red en términos dinámicos como la demanda de servicios, el tipo de servicios, la simultaneidad de los servicios invocados, el patrón de invocación de los servicios, etc.

La priorización de Indicadores de Negocio está fuera del alcance de este trabajo de Tesis. En este trabajo de Tesis hemos optado por encontrar los mejores valores para las políticas de gestión que mantengan un equilibrio en los tres indicadores de negocio, a tal medida que exista un compromiso entre la calidad de servicio y los indicadores, considerando los aspectos económicos de los proveedores de servicios.

Para solucionar esta problemática de encontrar las políticas que satisfagan los tres indicadores de negocio, hemos optado en utilizar algoritmos de optimización. *Optimizar* se define como “*el procedimiento de encontrar y comparar soluciones factibles hasta que una solución no pueda ser mejorada*” [9]. Las soluciones son denominadas aceptables en términos de los objetivos.

Aunque el espacio de búsqueda de las políticas de gestión es muy grande (políticas SR , TCL AC_{th} con valores comprendidos entre diversos valores mínimos y máximos) existen algoritmos de optimización que han sido reportados en la literatura para resolver este tipo de problemas. Por otra parte, la evaluación de los Indicadores de Negocio se realiza en tiempo real considerando las condiciones de la red y su comportamiento en términos dinámicos como la demanda de servicios, el

tipo de servicios, la simultaneidad de los servicios invocados, el patrón de invocación de los servicios, etc, lo cual permite encontrar valores con mucho más exactitud y precisión dentro del espacio de búsqueda, que al realizar los cálculos manualmente como en [12], los cuales no garantizan que se encuentren los valores ideales para las políticas de gestión.

En esta sección de la Tesis se describe el algoritmo que se ha definido para optimizar los indicadores de negocio en redes cognitivas de próxima generación el cual es resuelto como un problema de optimización multiobjetivo.

3.3.1 Uso de modelos surrogados para resolver problemas de optimización multiobjetivo en redes cognitivas de próxima generación

Tal como se mencionó anteriormente, esta Tesis se enfoca en resolver la problemática de optimizar los Indicadores de Negocio de modo que un proveedor de redes y servicios pueda elegir el área de operación que mejor le satisfaga tomando en cuenta sus intereses y tomando en cuenta que todos los Indicadores de Negocio definidos en la solución tienen la misma prioridad. Hasta el momento no existe reportado ningún trabajo en la literatura que haya abordado la optimización de los Indicadores de Negocio con este énfasis en el contexto de redes de servicios diferenciados con calidad de servicio como el abordado en este trabajo de Tesis. Es por ello que el enfoque que se plantea en esta Tesis, contribuye en gran manera al estado del arte. En este trabajo de Tesis se aborda el problema de optimizar los indicadores de negocio, optimización multiobjetivo que involucra tres objetivos.

Los Algoritmos Evolutivos (AEs) son metaheurísticas bioinspiradas que han sido exitosamente utilizadas para resolver problemas de optimización multiobjetivo (MOPs). Cuando estos problemas son computacionalmente costosos, incluso estas metaheurísticas pueden resultar inoperantes. Por lo tanto se necesitan estrategias adicionales con el fin de acelerar la convergencia de los AEs.

Cabe señalar que se ha utilizado la plataforma de desarrollo de redes especializada OPNET Modeler [1] para evaluar las funciones objetivo. La herramienta de simulación OPNET Modeler es capaz de simular una gran variedad de redes, además integra todas las capas del modelo TCP/IP y para nuestro fin, incorpora todas las especificaciones técnicas para proporcionar calidad de servicio

dentro de un dominio DiffServ. Esta plataforma se ha empleado para validar nuestra solución (descrito posteriormente). Sin embargo, nos hemos enfrentado con la problemática que al evaluar las funciones objetivo, el tiempo es sumamente costoso. Esto se debe a que el espacio de búsqueda de las políticas de gestión es muy grande (políticas SR , TCL , AC_{th} con valores comprendidos entre diversos valores mínimos y máximos), aunado a la facilidad que OPNET otorga para simular las condiciones de la red y su comportamiento en términos dinámicos como la demanda de servicios, el tipo de servicios, la simultaneidad de los servicios invocados, el patrón de invocación de los servicios, etc. Además, el comportamiento que simula OPNET de toda la pila de protocolos TCP/IP que tal como sucede en redes reales hace que la evaluación del desempeño de la red y de los objetivos de negocio sea muy eficiente pero muy costosa.

Debido a los tiempos excesivos de las funciones objetivo que se tardaría al aplicar AEs, hacen prohibitivo el uso de cualquier técnica de optimización [5].

La evaluación de nuestros objetivos de negocio en estas condiciones de simulación tan especializada con nuestra plataforma OPNET demanda técnicas de optimización eficientes. Inicialmente se experimentó con algoritmos evolutivos NSGA-II y SPEA obteniendo resultados en los cuales el tiempo de evaluación de los Indicadores de Negocio eran demasiados excesivos. Cabe mencionar que estos tipos de algoritmos trabajan con poblaciones, de tal modo que como mínimo el tamaño de una población está compuesto de 100 individuos y el número de generaciones máximas permitidas es de 250 generaciones, considerando 31 ejecuciones. Por otra parte, el tiempo de simulación en la plataforma OPNET al evaluar los Indicadores de Negocio es de 10 minutos.

El tiempo estimado al optimizar los Indicadores de Negocio aplicando estos tipos de algoritmos es el siguiente:

$$\text{Total de evaluaciones} = 7,500,000$$

$$\text{Tiempo de ejecución del algoritmo} = 5,208 \text{ días}$$

Por lo que el tiempo de ejecución estimado al aplicar algoritmos evolutivos como NSGA-II y SPEA sería de 5,208 días aproximadamente. Cabe mencionar que si se está considerando la utilización de algún otro algoritmo evolutivo teniendo los mismos valores en la configuración de los parámetros

mencionados anteriormente, el tiempo de ejecución sería aproximadamente el mismo. Por lo tanto, esto limita su aplicación y justifica el hacer prohibitivo el uso de estos algoritmos.

Por las proyecciones de evaluación para las ejecuciones necesarias, se determinó que los algoritmos evolutivos evaluados anteriormente son prohibitivos. Los modelos surrogados y las implementaciones paralelas son estrategias para minimizar el tiempo de respuesta ante esta clase de problemas.

Por una parte, las implementaciones paralelas se han popularizado como un mecanismo para mejorar la eficiencia de los AEs. Dividiendo la población entre varios elementos de procesamiento, los Algoritmos Evolutivos Paralelos (pEAs, por sus siglas en Inglés) permiten hallar resultados de alta calidad en tiempos razonables para problemas difíciles de resolver o de grandes dimensiones.

OPNET no soporta paralelización y por ello esta Tesis, no toma en cuenta la paralelización de AE's. Esta razón hace evidente a un problema complicado, por lo que hemos decidido excluir estos tipos de algoritmos.

Otra alternativa para minimizar el tiempo de respuesta ante esta clase de problemas son los modelos surrogados o metamodelos. Un metamodelo es una aproximación de la función original para construir modelos simples y de menor costo computacional [5]. Por todo lo anterior, y para resolver nuestra problemática, hemos optado en recurrir a los modelos surrogados.

3.3.2 Aplicación de los modelos surrogados para la optimización de los indicadores de negocio en redes cognitivas de próxima generación

Como se ha mencionado previamente, para resolver el problema propuesto hemos utilizado un modelo surrogado [5]. Para ello se propone el algoritmo de optimización multiobjetivo ParEGO, el cual toma los valores cuantitativos de los Indicadores de Negocio producidos por el *Evaluador de Indicadores de Negocio*, y el *módulo de Optimización de Indicadores de Negocio* realiza la ejecución del algoritmo de optimización ParEGO.

El objetivo de nuestro problema es encontrar las políticas de gestión que optimice los Indicadores de Negocio. En términos prácticos, el problema consiste en la minimización de 3 funciones objetivo que representan a cada indicador de negocio: *Satisfacción del servicio*, *Degradación del servicio* y

Pérdida por rechazo de invocaciones.

A continuación describimos la forma como se representan las soluciones iniciales y los parámetros generales que consideraremos en el algoritmo ParEGO empleado en esta solución propuesta. Para ello emplearemos la topología y los parámetros mostrados en la figura 3.5.

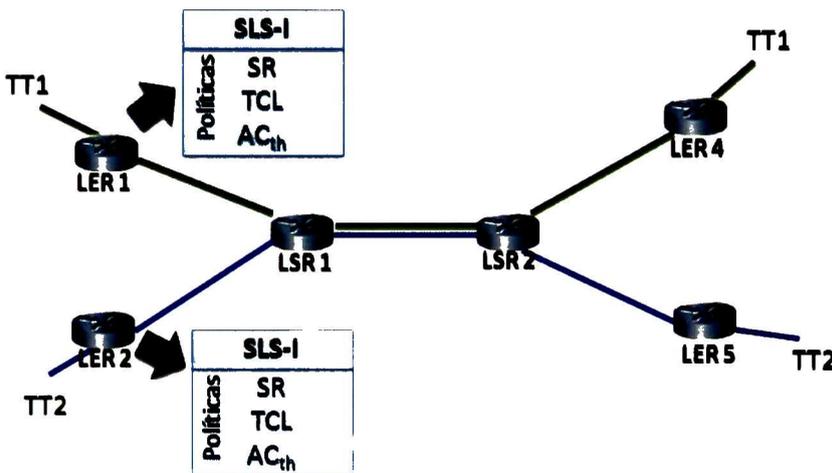


Figura 3.5: Topología de la red y parámetros de políticas SLS-I.

Soluciones iniciales. Como queremos encontrar los valores mínimos para los indicadores de negocio, es necesario definir cómo representaremos cada solución a la función a optimizar, en este caso consideramos cada solución como el conjunto de valores de las variables de control (ver figura 3.5): TCL_{TT1} , AC_{thTT1} , SR_{TT1} , TCL_{TT2} , AC_{thTT2} y SR_{TT2} , de tal modo que una solución esta compuesta por un conjunto de seis valores cuya configuración representa una solución posible para nuestro problema. Evidentemente, si la topología fuera diferente, es decir, si tuviera mas políticas o más TTs, el conjunto de variables sería correspondiente con dicha topología.

Inicialización de soluciones. Las soluciones iniciales se generan mediante *hipercubo Latino*, técnica de muestreo con la propiedad de generar muestras aleatorias [5] con un número inicial de

soluciones $11d - 1$, donde d representa la dimensión de la función a optimizar.

En el algoritmo 4 se muestra el pseudocódigo de ParEGO y los métodos que fueron incorporados en el algoritmo para evaluar los Indicadores de Negocio en la solución propuesta con la plataforma especializada.

Algoritmo 4 pseudocódigo: ParEGO implementado en el módulo Optimización de Indicadores de Negocio.

DEFINE ParEGO($f, d, \text{max_iters}, k, s$)

```

1:  $xpop[ ] \leftarrow \text{HiperCuboLatino}(d)$ 
2: for each  $i$  in 1 to  $11d - 1$  do
3:   Configurar Opnet con los valores del individuo
4:    $zpop[i] \leftarrow \text{Evaluar}(xpop[i], f)$ 
5: while  $11d - 1 < \text{max\_iters}$  do
6:    $\lambda \leftarrow \text{NEWLAMBDA}(k, s)$ 
7:    $model \leftarrow \text{DACE}(xpop[i], zpop[ ], \lambda)$ 
8:    $xnew \leftarrow \text{AlgEvol}(model, xpop[ ])$ 
9:    $xpop[ ] \leftarrow xpop[ ] \cup \{xnew\}$ 
10:  Configurar Opnet con los valores de  $xnew$ 
11:   $znew[ ] \leftarrow \text{Evaluar}(xnew, f)$ 
12:   $zpop[ ] \leftarrow zpop[ ] \cup \{znew\}$ 

```

END

En un inicio se genera un conjunto de soluciones mediante *hipercubo latino* con un tamaño de $11d - 1$ soluciones. Para el ejemplo que nos concierne de la figura 3.5 la dimensión de este problema es de 6 ($d = 6$), que corresponde al número de variables involucradas por individuo. El espacio de búsqueda se encuentra en los rangos de los valores mínimos y máximos de las políticas TCL , SR y AC_{th} para garantizar calidad de servicio (QoS); estos rangos están definidos por el problema, topología, número de SLAs, dimensionamiento, etc. a resolver tal como se ejemplificará en la Sección de validación (ver Capítulo 4). Una vez inicializados todos los valores que se requieren, el modelo de *Optimización de Indicadores de Negocio* de la arquitectura evalúa los $11d - 1$ soluciones que proporciona el *Evaluador de Indicadores de Negocio* de la Arquitectura propuesta. Cabe mencionar que antes de evaluar cada solución se configuran las políticas de gestión para cada una de las variables que pertenecen a dicha solución. Una vez inicializada y evaluada la solución se realiza lo siguiente

para las generaciones subsecuentes, hasta que se cumpla el número máximo de iteraciones requerido para las funciones a optimizar. Después de evaluar las soluciones, se ajustan los parámetros del modelo DACE (Design and Analysis of Computer Experiments - DACE, por sus siglas en Inglés) por medio de la *máxima verosimilitud*. El modelo DACE consiste en predecir un nuevo punto que no ha sido evaluado, además calcula el error estándar del predictor. Posteriormente para generar una nueva solución a evaluar, se busca la solución que maximiza *la mejora esperada* [8], mediante un algoritmo genético simple. Este algoritmo no sólo elige la solución que el modelo predice que reducirá al mínimo el valor. Por el contrario, de forma automática balancea la explotación y exploración. A continuación se muestrea la función en dónde la mejora esperada fue maximizada, obteniendo una nueva solución (el nuevo punto) que es evaluada con la verdadera función costosa con ayuda del módulo *Evaluador de Indicadores de Negocio*. El modelo DACE es actualizado con esta nueva información, y la próxima solución se elige utilizando este modelo actualizado. Finalmente se obtiene un conjunto de vectores de objetivos no dominados. El flujo de ejecución del surrogado aplicado en nuestro trabajo de Tesis se muestra en la figura 3.6.

3.3.3 Conclusiones

La gestión de redes y servicios ha estado tradicionalmente enfocada a satisfacer y controlar aspectos de calidad de servicio, y últimamente a través del uso de políticas de gestión, es posible mejorar la autonomía en la toma de decisiones de una manera dinámica y en tiempo real. Sin embargo, este dinamismo en la autonomía de gestión está desacoplado del valor de negocio de los operadores de redes y servicios.

La arquitectura propuesta en este trabajo de Tesis avanza el estado del arte al proponer los medios necesarios para elevar las capacidades de los operadores hacia la optimización de Indicadores de Negocio con la finalidad de encontrar configuraciones de políticas de gestión capaces de encontrar un compromiso entre los Indicadores de Negocio considerados, a la vez de considerar parámetros de calidad de servicio en la optimización. Se ha partido de una arquitectura inicial (TEQUILA – Traffic Engineering for Quality of Service the Internet at Large Scale) la cual ha sido necesario aumentar con

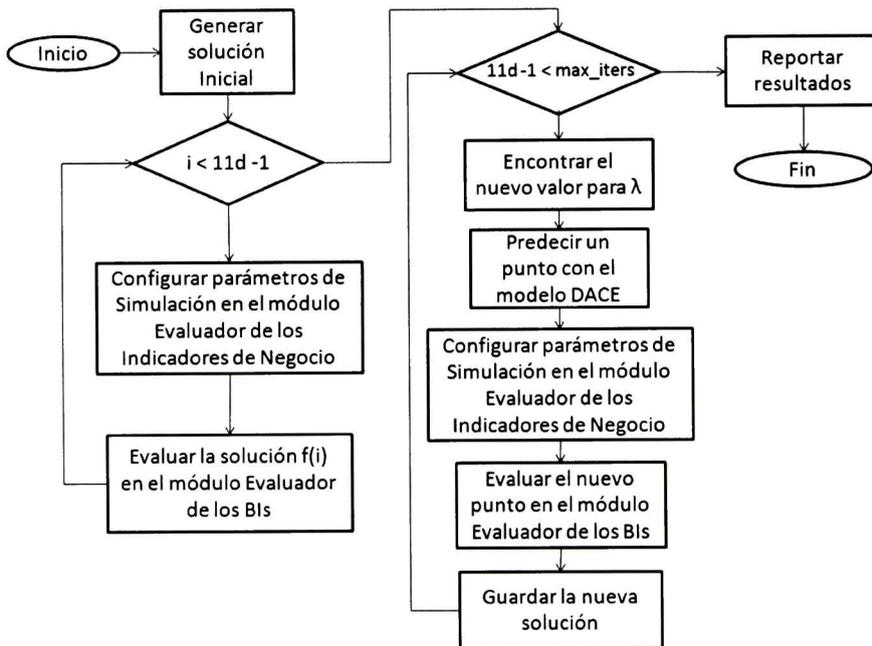


Figura 3.6: Flujo de ejecución de parEGO implementado en el módulo *Optimización de Indicadores de Negocio* .

funcionalidades de evaluación de Indicadores de Negocio y optimización de Indicadores de Negocio para lograr nuestros objetivos.

Respecto a la evaluación y optimización de indicadores de negocio, se puede concluir que un aspecto vital para llevar cabo la optimización es la formalización de Indicadores de Negocio y su cuantificación. Para ello, y complementaria a la definición de la arquitectura, se ha definido una metodología que ayuda a elevar el valor de negocio de los proveedores de servicios, la cual consiste en definir Indicadores de Negocio relevantes a servicios diferenciados con QoS, establecer la relación entre los Indicadores de Negocio con las políticas de gestión con QoS, establecer métricas para evaluar los Indicadores de Negocio, aplicar métodos de Optimización multiobjetivo para los Indicadores de Negocio.

Respecto a la metodología, hay dos puntos que merecen especial atención. Primeramente, cabe destacar que los Indicadores de Negocio que se definan, las políticas de gestión y las inter-relaciones que se definan, demandan un gran conocimiento del dominio de aplicación, en este caso, del contexto de servicios diferenciados (DiffServ) con calidad de servicio. Es posible que se pueda extrapolar la metodología y los elementos identificados en ella después de identificar las similitudes de este dominio de aplicación (DiffServ con calidad de servicio) con otro dominio de potencial aplicación.

Otro aspecto muy relevante que debemos resaltar es el que concierne a los procedimientos de optimización seleccionados. Aunque ciertamente como concepto, la optimización ha sido incluida en esta parte de diseño conceptual, la experimentación con diversos algoritmos de optimización ha dado pauta a definir algunos elementos de diseño de la solución propuesta. Debido a la naturaleza del problema abordado, y a la complejidad del espacio de búsqueda, es prácticamente inviable utilizar algoritmos evolutivos simples (e.g. NGS-II, SPEA) para resolver nuestro problema y ha sido necesario utilizar algoritmos surrogados, con los cuales se obtienen mejores resultados en la etapa de validación.

4

Validación Experimental

En este Capítulo se demuestra la viabilidad de la solución propuesta. Para ello se desarrolló un software de gestión el cual fue integrado en una plataforma especializada de simulación con la finalidad de validar los conceptos desarrollados en este trabajo de Tesis. Se ha ejecutado un escenario de prueba de concepto con funcionalidad servicios diferenciados de próxima generación. Para ello se ha utilizado la plataforma de simulación OPNET [1], la cual ha sido modificada explícitamente para cumplir con las especificaciones de diseño de la arquitectura propuesta.

Debido a que no hay ningún trabajo reportado en la literatura que aborde el problema de optimizar los Indicadores de Negocio en servicios diferenciados con calidad de servicio, no es posible realizar un estudio comparativo. Sin embargo, se ha encontrado en la literatura un método analítico de control de los indicadores de negocio mediante asignación de pesos de importancia [12], por lo que se ha tomado para comparar nuestro enfoque. En este capítulo se presentan los resultados obtenidos al llevar a cabo el proceso de optimización de los Indicadores de Negocio considerados en un escenario de servicios diferenciados con la solución propuesta.

4.1 Implementación del Enfoque propuesto

En esta sección se describe la implementación del enfoque propuesto, el cual está integrado por un Software de gestión de políticas, la plataforma Simulación de red OPNET modificada, el Software de evaluación de los Indicadores de Negocio (BIs) y el Mecanismo de optimización de los Indicadores de Negocio (BIs). En la figura 4.1 se muestra la configuración esquemática de todos estos elementos que mencionamos.

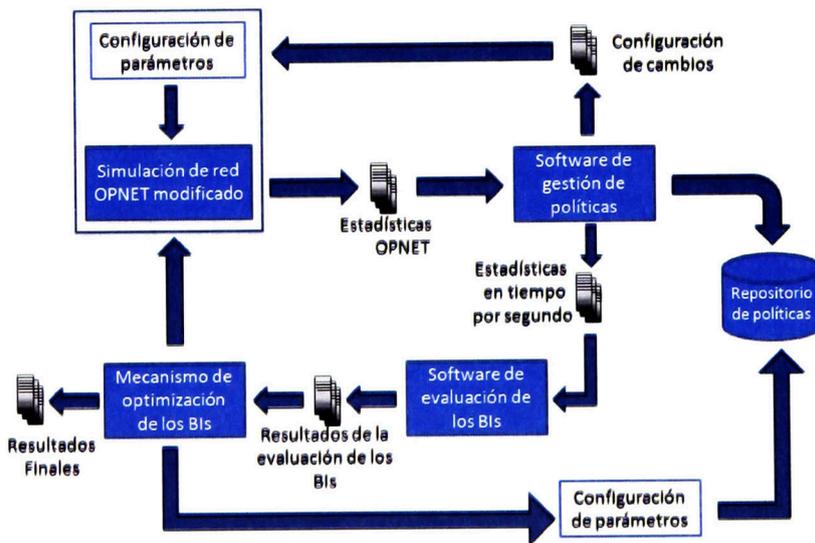


Figura 4.1: Integración del software de gestión, la plataforma de prueba, el software que evalúa los indicadores de negocio y el algoritmo que optimiza los indicadores.

El escenario de trabajo se desarrolló en la plataforma de Simulación de red OPNET modificada y el Software de gestión de políticas fue desarrollado en java. Para integrar ambos módulos fue necesario modificar la configuración de varios elementos en el simulador OPNET [1], con la finalidad entre otras cosas, de modificar la configuración de la red en tiempo real considerando las políticas SLS- que se han definido en la solución propuesta, para recopilar información de monitorización, también definidas para la solución propuesta. En particular estamos interesados en recolectar estadísticas del tráfico total agregado por clase de servicio dado en bits/seg.

El Software de gestión de políticas se ejecuta en paralelo al simulador, obtiene los datos de monitorización de la simulación, procesa el estado de la red, y ejecuta políticas procedentes del repositorio de políticas, considerando el estado actual de la red monitorizada. El software de gestión de políticas actúa de manera inteligente para analizar el comportamiento de la red dado que el software obtiene los datos del simulador referente a estadísticas de tráfico de las Troncales de Tráfico (Traffic Trunk – TT) durante la ejecución en tiempo de simulación y si hay cambios que aplicar, el simulador carga la nueva configuración de los parámetros y los aplica a la configuración de los routers frontera que corresponda según lo indiquen las acciones de las políticas que correspondan.

El Software de evaluación de los BIs es un software en java que se encarga de medir cada indicador de negocio respecto al comportamiento de la red. Este software analiza los archivos de salida proporcionado por el software de gestión y de acuerdo a las métricas descritas en la sección 3.2.3 obtiene los valores correspondientes para cada indicador, los resultados obtenidos los almacena en archivos de texto.

Los resultados del software de evaluación de Indicadores de Negocio son introducidos al Mecanismo de Optimización de BIs. Este último incluye una implementación del algoritmo de optimización ParEGO, el cual fue desarrollado en código c++, y el cual es el encargado de optimizar los indicadores de negocio. Sin embargo, para que esto sea posible el Mecanismo de Optimización implementa internamente un módulo que se encarga de configurar todos los parámetros necesarios para la correcta configuración del modelo de red. Cabe mencionar que el algoritmo ejecuta tanto el simulador OPNET Modeler como el software que evalúa los indicadores de negocio en cada de iteración.

4.2 Escenario de validación

En esta sección se definen tres escenarios de simulación, todos ellos con la misma topología con diferentes demandas de tráfico, considerados para realizar la experimentación. El objetivo es encontrar en todos ellos, las mejores configuraciones de políticas con los cuales exista un compromiso entre los valores de los tres Indicadores de Negocio considerados en nuestra solución. Los escenarios están basado en la tecnología DiffServ sobre MPLS, y han sido simulados en la plataforma de simulación

OPNET Modeler, capaz de soportar este tipo de tecnologías recreando un modelo lo más fiable posible a la realidad.

Consideremos un proveedor de redes y servicios que ofrece 14 SLAs. Cada SLA soporta una gran variedad de perfiles que se utilizan para obtener un patrón de tráfico de usuario. Cada perfil está conformado por un conjunto de aplicaciones. Para este escenario se definieron dos perfiles de servicios a los usuarios con las mismas aplicaciones (ver tabla 4.1):

Tabla 4.1: Perfiles de servicios.

Perfil medio	Perfil alto
http pesado FTP baja prioridad email pesado	http pesado FTP baja prioridad email pesado

El proveedor clasifica los servicios en calidad media y calidad alta, asignándolas a las clases de servicio AF11 y AF43, respectivamente. Se consideran 7 SLA de los servicios de calidad media con una tasa promedio de 3.43Mbps cada uno (tasa total promedio de 24 Mbps) y 7 SLA de los servicios de calidad alta con una tasa promedio de 1.714Mbps cada uno (tasa total promedio de 12Mbps).

En la tabla 4.2 se muestran los patrones de tráfico para cada una de las aplicaciones por conjunto de servicios. Estos patrones de tráfico se aplican para los tres escenarios de prueba.

Tabla 4.2: Patrones de tráfico para las aplicaciones por conjunto de servicios.

Aplicación	Variable	Distribución	Parám.	Instancias por servicio
http pesado	Page inter-arrival	Poisson	10 seg.	15 serv. calidad media
	Object Size	Constant	1KB	8 serv. calidad alta
FTP baja prioridad	Image	uniform	2kB-10kB	
	Inter-Request time	Poisson	8 seg.	15 serv. calidad media
email pesado	File Size	Poisson	500 kB	8 serv. calidad alta
	Inter-arrival time	Poisson	15 seg.	15 serv. calidad media
	E-mail Size	Poisson	4kB	8 serv. calidad alta

La topología de red se muestra en la figura 4.2. Desde LER1/LER4 son respectivamente los puntos de entrada/salida para los servicios de calidad media y de LER2/LER5 los de calidad alta. Los usuarios que pertenecen a los servicios de calidad media se asignan a los sitios del 1 al 7 y envían tráfico al

sitio 16. Los usuarios que pertenecen a los servicios de calidad alta se asignan a los sitios 8 al 14 y envían tráfico al sitio 17. El proveedor define los factores de satisfacción para cada clase de servicio: $Fctr_{AS} = 0.3$ y $Fctr_{FS} = 0.2$ para la clase de calidad media y $Fctr_{AS} = 0.4$ y $Fctr_{FS} = 0.3$ para la clase de calidad alta.

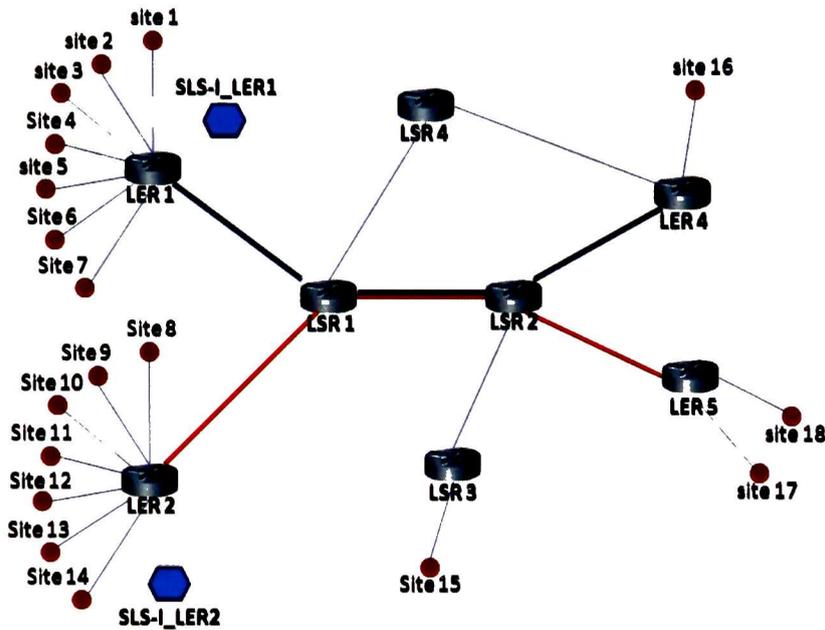


Figura 4.2: Topología de red para el análisis experimental

Tal como se mencionó anteriormente, se han definido tres escenarios de prueba, cuya diferencia entre sí es el patrón de invocación de los servicios y en consecuencia, el volumen y la manera en como inyecta el tráfico por los clientes en cada uno de los escenarios. Las especificaciones de cada escenario será descrito a continuación.

1. **Escenario 1.** En este escenario los servicios de calidad media se invoca cada 20 segundos y se mantienen activos durante 70 segundos. Los servicios de calidad alta se invoca cada 40 segundos y permanecen activos durante 80 segundos. El tiempo de inicio en que los sitios

comienzan a invocar los servicios es a los 20 segundos. El número de veces que los perfiles de servicio se repiten toma un valor de 8 para los servicios de calidad media y de 5 para los servicios de calidad alta.

2. **Escenario 2.** Para este otro escenario, los servicios de calidad media se invoca cada 10 segundos y se mantienen activos durante 70 segundos. Los servicios de calidad alta se invoca cada 20 segundos y permanecen activos durante 80 segundos. El tiempo de inicio en que los sitios comienzan a invocar los servicios es a los 20 segundos. El número de veces que los perfiles de servicio se repiten toma un valor de 18 para los servicios de calidad media y 9 para los servicios de calidad alta. Cabe señalar que este escenario maneja el doble de tráfico que el escenario 1.
3. **Escenario 3.** Para este escenario, las invocaciones a los servicios se hacen en diferentes tiempos. El sitio 1 invoca los servicios a los 20 segundos, el sitio 2 en el tiempo 25, el sitio 3 en el tiempo 30, el sitio 4 en el tiempo 35, el sitio 5 en el tiempo 30, el sitios 6 en el tiempo 20 y el sitio 7 en el tiempo 35 y se mantienen activos durante 70 segundos. Los servicios de calidad alta se invoca para el sitio 8 a los 20 segundos, el sitio 9 en el tiempo 30, el sitio 10 en el tiempo 20, el sitio 11 en el tiempo 25, el sitio 12 en el tiempo 30, el sitios 13 en el tiempo 25 y el sitio 14 en el tiempo 35 y permanecen activos durante 80 segundos. El tiempo de inicio en que los sitios comienzan a invocar los servicios es a partir del los 20 segundos. El número de veces que los perfiles de servicio se repiten toma un valor de 9 para los servicios de calidad media y 6 para los servicios de calidad alta.

El tiempo de simulación considerado en todos los escenarios de prueba fueron de 5 minutos.

4.2.1 Dimensionamiento de Red

De una manera similar que con la arquitectura TEQUILA (Traffic Engineering for Quality of Service for the Internet at Large Scale), en la solución propuesta la información descrita en la sección anterior, pasa a un proceso de dimensionamiento [12] resultando en la creación de dos TT con el siguiente formato:

$$TT = [\text{ingreso/egreso}, QC, RAB_{TT}] \quad (4.1)$$

donde

$$TT1 = [LER1/LER4, AF11, RAB_{TT1}] \quad (4.2)$$

$$TT2 = [LER2/LER5, AF43, RAB_{TT2}] \quad (4.3)$$

Se calculan los valores de la disponibilidad de recursos del buffer $RAB[R_{min}^a, R_{min}^w, R_{max}]$ para cada clase de servicio. Los valores de R_{min}^a , R_{min}^w y R_{max} consideran los factores de satisfacción. Teniendo los factores de satisfacción es posible calcular las demandas de tráfico para los servicios de calidad media y alta: TD_{min} : Demanda de tráfico mínima y TD_{max} : Demanda de tráfico máxima, las cuales se calculan con las siguientes expresiones:

$$TD_{min} = (1 - Fctr_{AS})(\text{tasa total promedio}) \quad (4.4)$$

$$TD_{max} = (1 + Fctr_{FS})(\text{tasa total promedio}) \quad (4.5)$$

La demanda de tráfico mínima y máxima para cada clase de servicio son:

- "calidad media":

$$TD_{minTT1} = (1 - 0.3)(24Mbps) = 16.8Mbps = R_{minTT1}^a \quad (4.6)$$

$$TD_{maxTT1} = (1 + 0.2)(24Mbps) = 28.8Mbps \quad (4.7)$$

- "calidad alta":

$$TD_{minTT2} = (1 - 0.4)(12Mbps) = 7.2Mbps = R_{minTT2}^a \quad (4.8)$$

$$TD_{maxTT2} = (1 + 0.3)(12Mbps) = 15.6Mbps \quad (4.9)$$

Para determinar R_{min}^w calculamos los recursos que sobran después de haber calculado R_{min}^a . El enlace físico que comparten ambas TTs cuenta con una capacidad de 34.36Mbps, representa los recursos máximos disponibles a lo largo de la ruta, para ambas TT. En base a estos datos, se calculan los recursos que sobran sumando las R_{min}^a de las TTs, valor que se le resta a la capacidad del enlace que se comparte. En este caso, los recursos que sobran son 10.37Mbps. Estos recursos se dividen entre TT1 y TT2 en proporción a su valor TD_{max} , esto es, 6.9Mbps para TT1 y 3.5Mbps. En consecuencia, $R_{minTT1}^w = 6.9Mbps + R_{minTT1}^a = 23.7Mbps$. Similar para TT2, $R_{minTT2}^w = 3.5Mbps + R_{minTT2}^a = 10.6Mbps$.

Finalmente, para calcular R_{max} para una TT se realiza restando la suma R_{min}^a de las TT que comparten los recursos, de la capacidad del enlace que se comparte, esto es, $R_{maxTT1} = 34.36Mbps - 7.2Mbps = 27.1Mbps$. Similar para TT2, $R_{maxTT2} = 34.36Mbps - 16.8Mbps = 17.5Mbps$. Para mayor detalle ver en [12].

El buffer RAB obtenido para TT1 y TT2 son:

$$RAB_{TT1} = [16.8Mbps, 23.7Mbps, 27.1Mbps]$$

$$RAB_{TT2} = [7.2Mbps, 10.6Mbps, 17.5Mbps]$$

Los valores de R_{min}^a , R_{min}^w y R_{max} son significativamente importantes para la aplicación de las políticas SLS-I orientadas al negocio, y más aún para nuestra tarea de optimización puesto que éstos vienen a determinar un aspecto importante en el espacio, muestra de las políticas TCL , SR y AC_{th} del algoritmo de optimización, tal y como se describe a continuación

4.2.2 Rangos propuestos de las políticas de gestión

Los valores posibles para las políticas de gestión, se encuentran dentro de los rangos que se calcularon en la sección anterior.

Para la TT1 se tienen los siguientes rangos de valores:

- $TCL \in [R_{min}^a, R_{max}] = [16.8Mbps, 25.7Mbps]$
- $AC \in [TCL, R_{max}] = [TCL, 27.1Mbps]$
- $SR \in [R_{min}^a, R_{max}] = [16.8Mbps, 27.1Mbps]$

Para la TT2 se tienen los siguientes rangos de valores:

- $TCL \in [R_{min}^a, R_{max}] = [7.2Mbps, 16.6Mbps]$,
- $AC \in [TCL, R_{max}] = [TCL, 17.5Mbps]$
- $SR \in [R_{min}^a, R_{max}] = [7.2Mbps, 17.5Mbps]$

Cabe mencionar, que a efectos prácticos y por motivos del contexto de aplicación hemos aplicado el 95% de R_{max} al rango superior para la política TCL. Esto se hace con la finalidad de lograr ejercer las acciones a tomar cuando el umbral sea excedido del valor especificado. Si el umbral de la política TCL se fijara con el valor máximo que corresponde a R_{max} , no sería posible que el umbral sea sobrepasado debido a que el tráfico que entra en la red llega y que hasta ese límite estaría en el mismo nivel que el TCL en ese caso particular..

Por otro lado, en el caso del SR, se ha hecho una aproximación del valor que la política tomaría con respecto a los valores definidos en el simulador OPNET, esto se hace con la finalidad de identificar los valores dentro del simulador, ya que el simulador tratará de buscar tal valor para ejercer la acción a tomar. Se han definido 200 valores para cada clase de servicio de nuestro ejemplo dentro del simulador. El valor que se le asigne al SR será aproximado a uno de los valores que fueron definidos en OPNET.

Por último, los tres objetivos que corresponden a los indicadores de negocio están normalizados en el rango $[0, 1]$.

4.2.3 Configuración de parámetros del algoritmo ParEGO

Los experimentos fueron ejecutados 20 veces para cada escenario de manera independiente. Los parámetros del algoritmo utilizado fueron: un conjunto de soluciones iniciales de $11d-1=65$ para todos los casos, donde $d=6$ durante 86 iteraciones. Los parámetros del algoritmo utilizado en ParEGO para buscar la solución que maximiza la mejora esperada se implementa de la siguiente manera: El tamaño de la población es de 20 individuos, una probabilidad de cruce de 0.2, una probabilidad de mutación de $1/\text{NumV}$, donde NumV es el número de variables del problema. En cada caso se realizaron 100,000 generaciones.

4.2.4 Resultados y Análisis

En esta sección se presentan los resultados de la optimización de los indicadores de negocio para los tres escenarios de prueba y un análisis de dichos resultados.

En la figura 4.3 se muestra el frente de Pareto obtenido de las soluciones del escenario de prueba 1. En él se muestra el conjunto de soluciones óptimas para los Indicadores de Negocio.

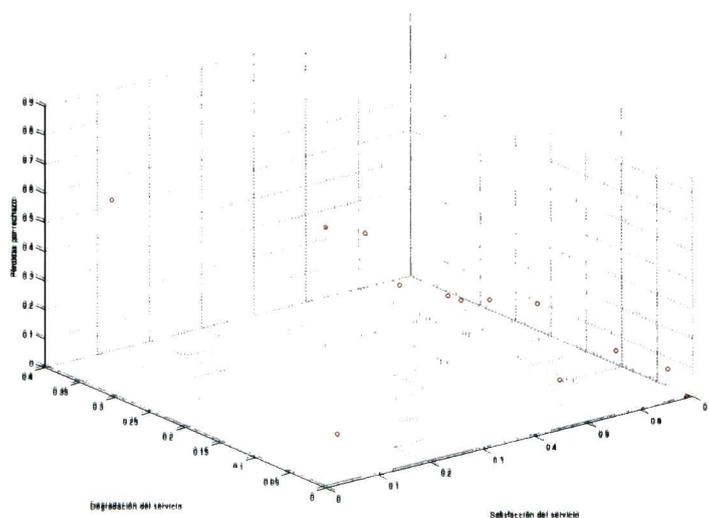


Figura 4.3: Frente de Pareto del escenario de prueba 1

En la figura 4.4 se muestra el frente de Pareto obtenido de las soluciones del escenario de prueba 2. En él se muestra el conjunto de soluciones óptimas para los Indicadores de Negocio para este tipo de topología de red y patrones de tráfico.

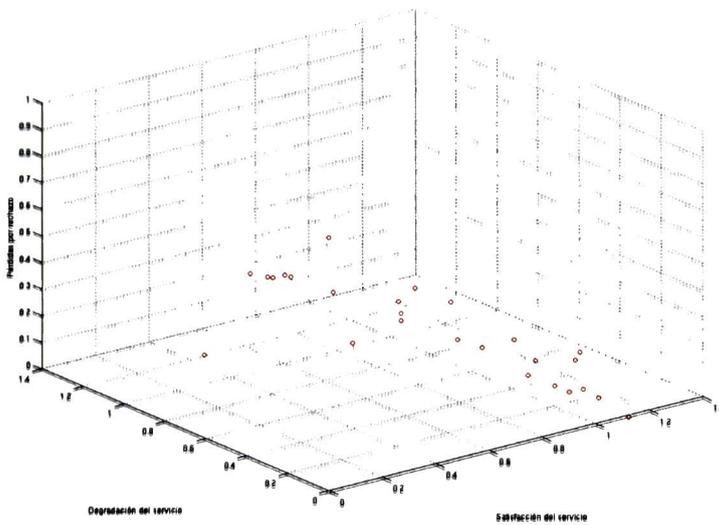


Figura 4.4: Frente de Pareto del escenario de prueba 2

En la figura 4.5 se muestra el frente de Pareto obtenido de las soluciones del escenario de prueba 3. Se observa que se obtienen más soluciones que los escenarios anteriores, debido que la invocación de los servicios se realiza de manera desfazada y esto hace que el comportamiento de la red cambie en gran manera.

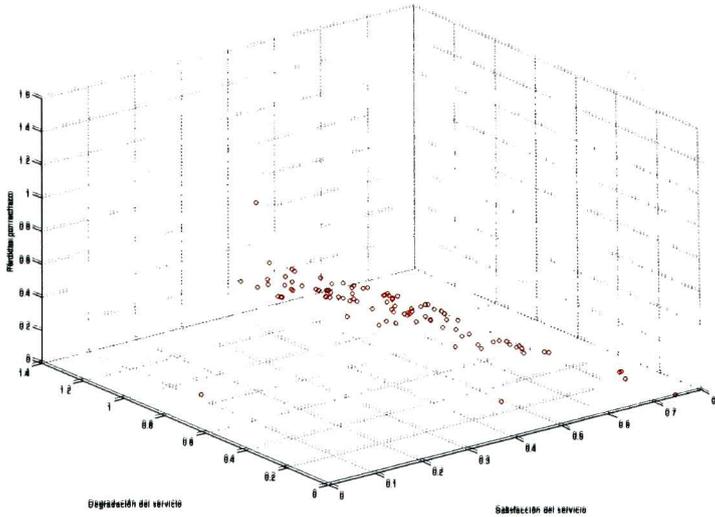


Figura 4.5: Frente de Pareto del escenario de prueba 3

Con la finalidad de encontrar las mejores soluciones para los indicadores de negocio, se procede a realizar un filtrado del conjunto de soluciones no dominadas encontradas en cada ejecución, donde las soluciones cuyos vectores de objetivos no dominados con ranking uno (soluciones en primera posición del frente de Pareto), conforman el frente de Pareto. Posteriormente, se aplica la métrica de distancia generacional del frente de Pareto obtenido y el conjunto de soluciones no dominadas de cada ejecución. Esto se realiza para cada escenario de prueba. Distancia generacional se define como la manera de medir la cercanía de las soluciones encontradas por el algoritmo, con respecto a las soluciones que se encuentran en el verdadero frente de Pareto [9].

La tabla 4.3 muestra los resultados de la métrica distancia generacional para los tres escenarios. Estos resultados corresponden a la mejor, peor, media y desviación estándar de las 20 ejecuciones realizadas de cada escenario.

En el Escenario 1 se observa que la mejor solución tuvo una cercanía al frente de Pareto con una distancia de 0.027822, para el Escenario 2 se obtuvo una distancia de 0.035614 y en el Escenario 3 la distancia fue de 0.010019.

Para comparar estos experimentos hemos propuesto una solución que se obtiene al asignarle pesos

Tabla 4.3: Resultados de la distancia generacional de cada escenario.

Escenario	Mejor	Peor	Media	Desviación Estándar
Escenario 1	0.027822	0.079366	0.060070	0.012437
Escenario 2	0.035614	0.129780	0.079115	0.022706
Escenario 3	0.010019	0.020346	0.015635	0.002530

de importancia a los indicadores de negocio, método tomado de la literatura [12]. Hemos asignado pesos con la misma prioridad a los tres indicadores de negocio (Satisfacción del servicio, Degradación del servicio y Pérdida por rechazo), donde $w_3=w_2=w_4=0.5$, respectivamente, de tal manera que los tres indicadores estén equilibrados. De acuerdo a la prioridad asignada, se derivan las políticas de gestión. Los valores que se obtuvieron en cada escenario al ejecutar la simulación en OPNET con las correspondientes políticas se muestran en la tabla 4.4.

Tabla 4.4: Resultados obtenidos al evaluar los indicadores de negocio mediante asignación de pesos de importancia.

Indicador de negocio	Escenario 1	Escenario 2	Escenario 3
Satisfacción del servicio	0.32327	0.74815	0.26927
Degradación del servicio	0.59682	1.16066	0.38774
Pérdida por rechazo	0.94444	0.91578	0.85714

Se calcula la distancia Euclidiana entre el punto propuesto y la solución más cercana al frente de Pareto, para cada escenario de prueba, de modo que mientras más reducida sea la distancia, mejor es el punto propuesto. Sin embargo, en la tabla 4.5 es posible apreciar que el punto propuesto es dominado incluso por la peor solución obtenida por la solución propuesta con el software implementado en este trabajo de Tesis con ParEGO, lo que significa que es dominada por todas las soluciones encontradas por el algoritmo. Por lo tanto, al comparar la solución propuesta con la peor solución, es significativamente mejor la peor solución que la solución propuesta, esta situación puede observarse para cada escenario de prueba.

Los resultados de dicho estudio demuestran que al optimizar los indicadores de negocio obtenemos un conjunto de soluciones con mejores valores para los indicadores de negocio con sus respectivas políticas de gestión, que al aplicar las ecuaciones analíticas dadas en [12]. Además, nuestro enfoque propuesto considera los tipos de aplicaciones, patrones de tráfico e invocación, demandas de tráfico,

Tabla 4.5: Distancia Euclidiana y Distancia generacional.

Escenario	distancia Euclidiana	Distancia generacional peor
Escenario 1	0.258213	0.079366
Escenario 2	0.326861	0.129780
Escenario 3	0.028511	0.020346

congestionamiento y entre otros aspectos de vital importancia para evaluar los indicadores de negocio, parámetros que no consideran en el enfoque propuesto en la literatura.

4.3 Conclusiones

La optimización de los indicadores de negocio en una red de comunicaciones es una tarea compleja, pues involucra una gran cantidad de factores que son necesarios evaluar. A pesar de la complejidad que existe, en este capítulo ha quedado de manifiesto que es posible optimizarlos y se ha dado una prueba de la viabilidad de los instrumentos necesarios para llevar esta promesa a la práctica. Se aplicó un algoritmo de optimización multiobjetivo que adopta el uso de los modelos surrogados, debido al costo computacional al evaluar los Indicadores de Negocio en la plataforma de simulación empleada para la validación de la solución propuesta. Después de una serie de experimentos podemos llegar a la conclusión que al optimizar los Indicadores de Negocio un proveedor de redes y servicios puede tener la facilidad de elegir el conjunto de soluciones encontradas que mejor refleje el compromiso entre objetivos de negocio de su empresa. De esta manera los indicadores de negocio y las políticas de gestión evolucionan en función del beneficio económico que obtienen los proveedores y están correlacionadas con aspectos dinámicos como la demanda de servicios, el tipo de servicios, la simultaneidad de los servicios invocados, el patrón de invocación de los servicios, etc. Cabe destacar que como resultado del análisis de la validación experimental, los resultados demuestran que al optimizar los indicadores de negocio con la solución propuesta en este trabajo de Tesis, se obtienen soluciones con mejores valores para los Indicadores de Negocio con sus respectivas políticas de gestión, que al aplicar las ecuaciones analíticas existentes hasta hoy en la literatura para el dominio de aplicación abordado en este trabajo.

5

Conclusiones

Un proveedor de redes y servicios establece metas y objetivos para su negocio, de tal forma que mantenga un compromiso para ofrecer a los clientes calidad de servicio. Para cumplir tales metas y objetivos se definen los Indicadores de Negocio, de esta manera miden la situación en la que se encuentra el negocio, si realmente se están cumpliendo los objetivos y metas establecidas. Las políticas de gestión ayudan a lograr automatizar el comportamiento de la red tomando en cuenta los objetivos del proveedor.

En esta tesis se ha propuesto una arquitectura y una metodología de gestión que optimiza los Indicadores de Negocio, los cuales consideran tanto aspectos técnicos de la red como aspectos económicos de los proveedores de servicios. Para hacer esto posible, se han establecido métricas para tales indicadores y poder evaluarlos. Hoy en día no existen métricas que evalúen los Indicadores de Negocio e informen al proveedor de redes y servicios la situación de su negocio.

Con la optimización de los indicadores de negocio, un proveedor de redes y servicios puede elegir de un conjunto de soluciones la solución que mejor le proporcione aceptable para las políticas de gestión, de tal manera que ayuden a mantener un equilibrio en los tres indicadores de negocio y la calidad de servicio. La complejidad en el funcionamiento de las redes cognitivas de próxima generación

hace necesaria la utilización de algoritmos de optimización basado en modelos surrogados. Debido a que los indicadores de negocio son computacionalmente costosos de evaluar, ha sido necesario aplicar esta clase de algoritmos para minimizar el tiempo de respuesta.

De acuerdo con el estado del arte y los resultados obtenidos en las pruebas de concepto, se observa que al optimizar los indicadores de negocio ayuda a obtener mejores resultados, de tal manera que los proveedores de servicios podrán configurar la red acorde a los objetivos de negocio que se hayan establecido.

5.1 Contribuciones

Las principales contribuciones de este trabajo de tesis son las siguientes:

- Se contribuyó al estado del arte con una arquitectura de gestión que optimiza los indicadores de negocio en redes fijas de próxima generación, tal arquitectura considera aspectos de calidad de servicio y de negocio, además de que los objetivos de negocio y de calidad de servicio están totalmente influenciados por las condiciones dinámicas de utilización de servicios o de las condiciones de la red.
- Se propuso una metodología de gestión que es capaz de optimizar los indicadores de negocio utilizando gestión basada en políticas, de tal modo que ambos consideren aspectos tecno-económicos de los proveedores de servicios, aplicándola al contexto de servicios diferenciados de próxima generación con calidad de servicio.

5.2 Dificultades Técnicas

Durante el desarrollo de esta Tesis se presentaron diversas dificultades técnicas que se han resuelto para culminar favorablemente. Uno de los retos principales que abordamos en esta Tesis, fue buscar una métrica que midiera los Indicador de Negocio, considerando varios factores involucrados como los son número de usuario, tipo de servicio, invocaciones de servicio y otros aspectos relevantes.

Se tuvo la dificultad de realizar la integración de la plataforma de validación con el algoritmo propuesto en esta tesis, el software que evalúa los Indicadores de Negocio y el software de gestión de políticas, debido a que se tuvo que buscar un mecanismo de comunicación entre todos ellos.

Otro de los retos principales que enfrentamos fue el analizar diferentes algoritmos y considerar el que mejor resultados proporcionara en un tiempo de cómputo considerable, debido que el tiempo de evaluación de los Indicadores de Negocio era sumamente excesivo, lo que dió pauta a la prohibición de varios algoritmos propuestos en la literatura.

Aunque el tiempo de parejo en comparativa con los algoritmos evolutivos existentes en la literatura era mucho menor al momento de obtener resultados óptimos, fué necesario ejecutar el algoritmo en distintas máquinas para reducir el tiempo, lo que llevó a tener que realizar las configuraciones apropiadas en el simulador OPNET en conjunto con el algoritmo y demás archivos para nuestro fin.

5.3 Conclusiones

Al optimizar los Indicadores de Negocio un proveedor de redes y servicios puede tener la facilidad de elegir el conjunto de soluciones encontradas que mejor refleje el compromiso entre objetivos de negocio de su empresa. De esta manera los indicadores de negocio y las políticas de gestión evolucionan en función del beneficio económico que obtienen los proveedores y están correlacionadas con aspectos dinámicos como la demanda de servicios, el tipo de servicios, la simultaneidad de los servicios invocados, el patrón de invocación de los servicios, etc.

Al optimizar los Indicadores de Negocio con la solución propuesta en este trabajo de Tesis, se obtienen soluciones con mejores valores para los Indicadores de Negocio con sus respectivas políticas de gestión, que al aplicar las ecuaciones analíticas existentes hasta hoy en la literatura para el dominio de aplicación abordado en este trabajo.

5.4 Trabajo futuro

Debido a que la optimización de los indicadores de negocio se realizó para una cierta topología de red, como trabajo futuro se plantea optimizar los indicadores de negocio para servicios de próxima generación, analizando diferentes tipos de topologías de redes más complejas, esto permitirá que no sólo se encuentren las mejores soluciones para los indicadores de negocio de cierta topología de red, sino para un número n de topologías en redes fijas y encontrar la que proporcione mejores resultados.

Otro punto importante sería optimizar los indicadores de negocio para servicios de próxima generación con diferentes patrones de tráfico (invocaciones, tipos de servicios, aplicaciones, etc), en donde no únicamente tengan valores fijos estos patrones de datos, sino que puedan ser dinámicos.

Por último se propone optimizar los indicadores de negocio para servicios de próxima generación en redes inalámbricas. Esta propuesta es un tema muy amplio que involucra muchos parámetros por analizar, debido a que las redes inalámbricas funcionan muy distinto a las redes fijas. Otro aspecto es el de definir prioridades en la optimización, dado que en este trabajo se ha considerado que todos los indicadores de negocio tienen la misma prioridad.

A

Diagrama de clases

En este anexo se muestra el diagrama de clases correspondiente al software de gestión de políticas y al software de evaluación de los BIs. Se detalla el funcionamiento de los diferentes métodos correspondientes a cada software.

A.1 Diagrama de clases del software de gestión de políticas

Tal como se mencionó anteriormente, el software de gestión de políticas está en constante monitorización del estado de la red, actúa de manera autónoma sobre las acciones a llevar a cabo. El software fue desarrollado en netbeans 6.9.1. En la figura A.1 se muestran las clases que corresponden al software de gestión de políticas. La tabla A.1 muestra las clases más relevantes.

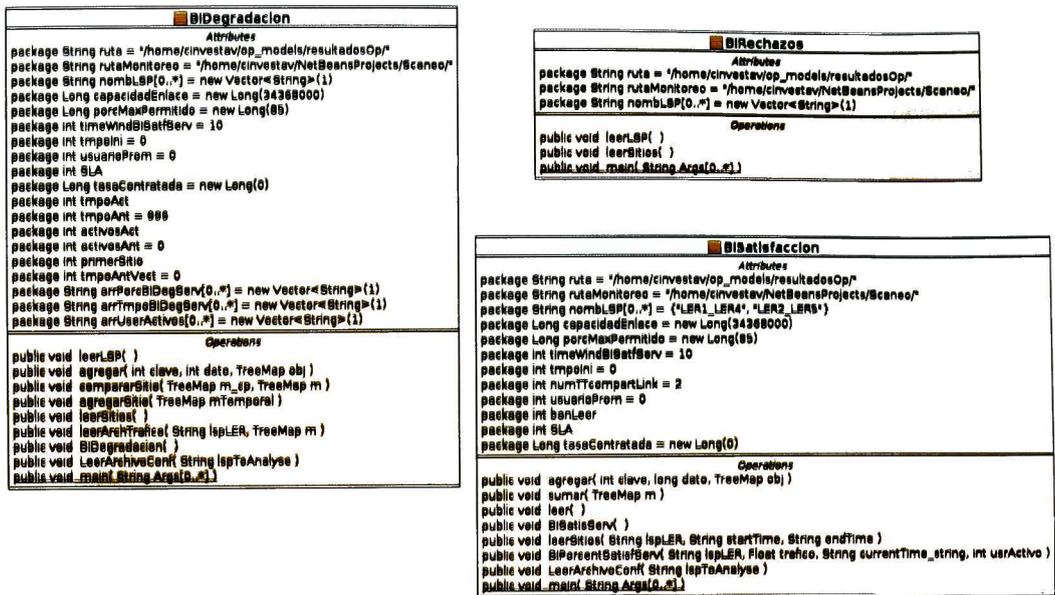


Figura A.1: Diagrama de clases del software de gestión de políticas

A.2 Diagrama de clases del software de evaluación de los BIs

El software de evaluación de los BIs fue desarrollado en netbeans 6.9.1 y su objetivo principal es evaluar los indicadores de negocio. Este software establece una relación con el software de gestión de políticas, tal que este último proporciona información relevante para que el software de evaluación de los BIs obtenga los valores correspondientes para cada indicador. La figura A.2 muestra el diagrama de clases del software implementado.

En la tabla A.2, A.3 y A.4 se describen los métodos principales de la clase BIDegradacion, BIRchazos y BISatisfaccion, respectivamente.

Nombre del método	Descripción
TraffRate	Esta clase se encarga de hacer cumplir las políticas cuando se ha sobrepasado el umbral. El mecanismo que utiliza es activando cada una de las políticas que corresponden a cierta clase de servicio. Si OPNET detecta activa las políticas, procede a realizar los cambios necesarios.
TrafficRateInPerLSP	Esta clase se encarga de leer las estadísticas del simulador OPNET a partir de un archivo de texto, transforma el tráfico que entra en la red en tiempo de simulación a tiempos por segundos. Además, analiza si no se ha superado el umbral que se ha fijado por alguna de las políticas de gestión, si esto ocurre toma la acción a considerar.

Tabla A.1: Descripción de métodos implementados en el software de gestión de políticas.

Nombre del método	Descripción
leerSitios	Este método contabiliza el número de sitios activos para cada clase de servicio.
leerArchTrafico	Este método se encarga de leer el tráfico por segundos proporcionado por el software de gestión de políticas y agrega los usuarios activos en cada segundo.
BIDegradacion	Calcula el porcentaje que corresponde a la degradación del servicio para cada clase de servicio, tomando en cuenta una ventana de tiempo de 10 segundos. Si la ventana se cumple procede a calcular el porcentaje, de lo contrario se reinicializa el tamaño de la ventana

Tabla A.2: Descripción de métodos implementados en la clase BIDegradacion.

Nombre del método	Descripción
leerSitios	Este método calcula el total de usuarios rechazados proporcionados por el simulador OPNET.

Tabla A.3: Descripción de métodos implementados en la clase BIRchazos.

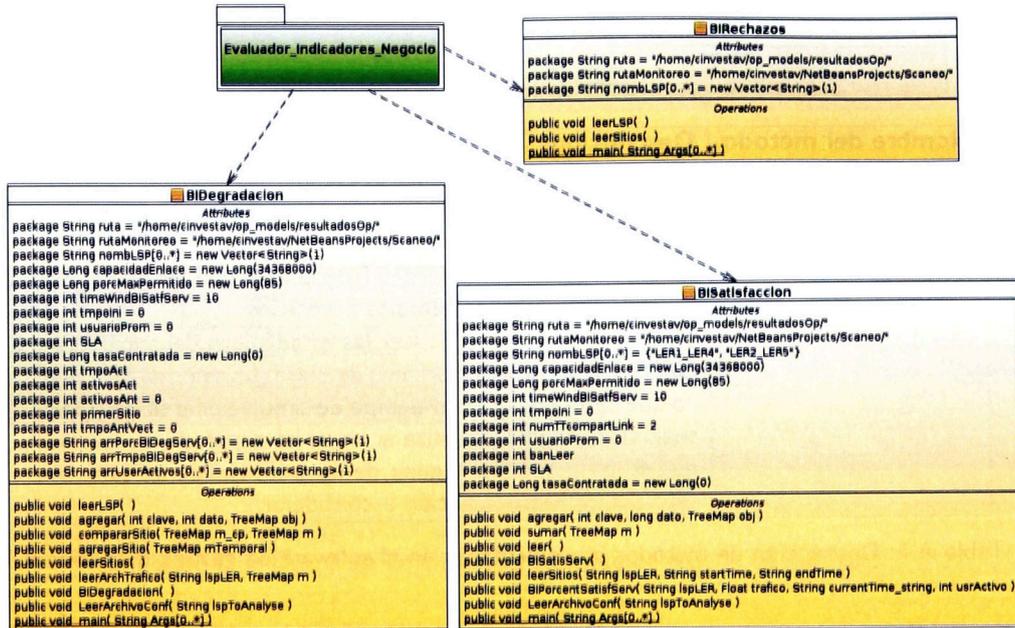


Figura A.2: Diagrama de clases del software de evaluación de los BIs

Nombre del método	Descripción
sumar	Este método es el encargado de sumar el tráfico total de todas las TT que comparten el mismo enlace. Como resultado se obtiene la utilización del enlace durante cada segundo.
BISatisServ	Este método compara si la capacidad del enlace compartido ha excedido al 85 % de su utilización. Si se ha excedido del 85 % se calcula el porcentaje del BI, de lo contrario el BI es igual a 1. La salida es un archivo con el porcentaje del BI correspondiente.
BIPorcetSatisfServ	En este método se realizan las operaciones necesarias para obtener el porcentaje que corresponde al indicador de negocio.

Tabla A.4: Descripción de métodos implementados en la clase BISatisfaccion.

B

Modificaciones en OPNET

Dado que el comportamiento original de OPNET no permite consultar los datos de la red en tiempo de ejecución y mucho menos modificar la configuración de la red en tiempo real, fue necesario modificar varios procesos en OPNET para lograr este comportamiento. En este apéndice se describen los procesos que fueron modificados en la plataforma de simulación OPNET Modeler.

B.1 Módulos de proceso modificados en OPNET Modeler

Opnet utiliza una estructura jerárquica en tres niveles: modelo de red, modelo de nodos y modelo de procesos. En este apéndice haremos referencia al modelo de procesos, el cual permite definir el comportamiento subyacente de los otros modelos.

Los modelos de proceso son máquinas de estados finitos, donde los estados son representados como iconos y las transiciones entre estados como líneas. Las operaciones que se realizan en cada estado o por una transición se describen en bloques de código en un lenguaje llamado Proto C. Estos estados y transiciones fueron modificados en los procesos que se mencionan a continuación.

B.1.1 Proceso mpls_mgr

El proceso *mpls_mgr* es modificado para recolectar la cantidad de tráfico dentro de la red en tiempo de simulación. En la figura B.1 se muestra el diagrama del tal proceso.

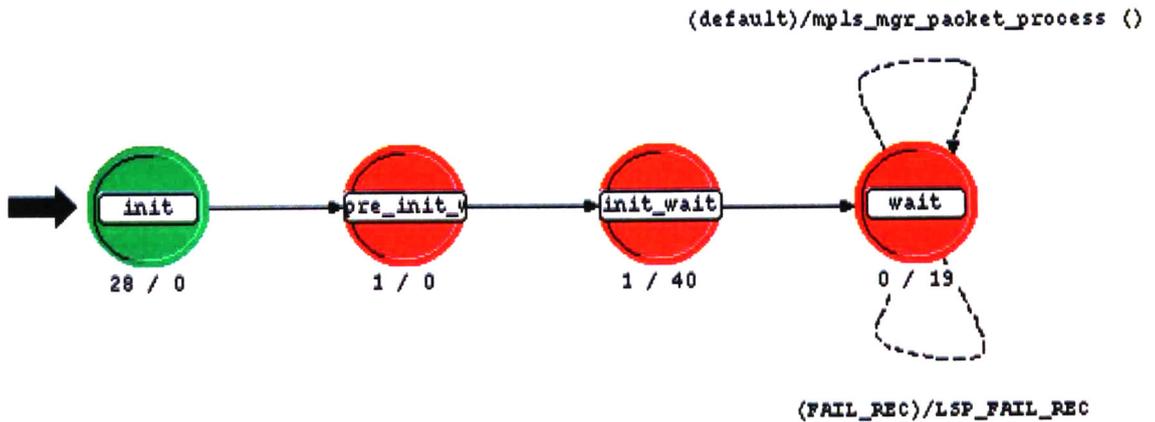


Figura B.1: Estados del proceso *mpls_mgr*

Este proceso necesita ser modificado en el *Function Block* con el siguiente código:

```
op_ima_obj_attr_get(state_ptr->nhlfe_ptr->lsp_objid, "name", annotation_str);
sprintf (file_str_2, "ruta/%s.in txt", annotation_str);
f2 = fopen(file_str_2, "ab");
if (f2 == NULL)
{
    op_sim_message ("f2 es NULL", "");
}

sprintf (annotation_str_2, "%s // Traffic Input bps-> %f // Time-> %f",
        annotation_str, pk_size, op_sim_time());

if (ip_print==0)
{
```

```

    sprintf ( annotation_str , "%s —> %s" , src_addr_str , dest_addr_str );
    op_sim_message ( annotation_str , "m5" );
    ip_print=1;
}
fprintf(f2," %s \n" ,annotation_str_2);
fclose(f2);

```

B.1.2 Proceso pl_vm

Este proceso es el encargado de estar interactuando con las políticas de gestión y de efectuar la nueva configuración en la red. Dado que en OPNET no existe tal proceso fue necesario crearlo. En la figura B.2 se muestran los estados de tal proceso.

Dentro del *Enter Executives* del estado *init* se agregó el siguiente código:

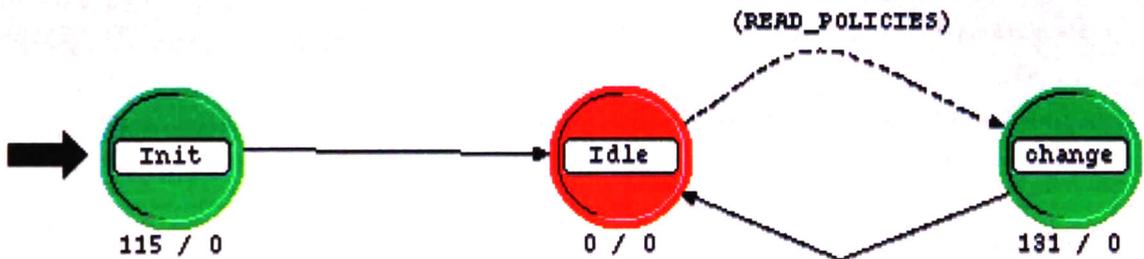


Figura B.2: Diagrama del proceso *pl_vm*

```

temporal1 = op_prg_mem_alloc(100);
temporal2 = op_prg_mem_alloc(100);
temporal3 = op_prg_mem_alloc(100);
temporal4 = op_prg_mem_alloc(100);
interface = op_prg_mem_alloc(100);
router    = op_prg_mem_alloc(100);

```

```

nomLSP      = op_prg_mem_alloc(100);
read_SLSI_Policy  = op_prg_mem_alloc(500);
policy_filename  = op_prg_mem_alloc(500);
site_index = 0;
invoke = 0;

/* Inicializar el numero de sitios configurados */
op_ima_obj_attr_get(op_id_self(), "Sites", &sites_comp_objid);
num_sites = op_topo_child_count (sites_comp_objid , OPC_OBJTYPE_GENERIC);

/*Guardando el numero de Sitios en un archivo llamado numSitios.txt*/
fSite=fopen("ruta/numSitios.txt","ab");
temporalSite = op_prg_mem_alloc(100);

/*Se obtiene la traffic trunk a la que pertenece el num. de sitio*/
op_ima_obj_attr_get(op_id_self(), "Router", router);
router_obj = op_id_from_name(0, OPC_OBJTYPE_NODE_FIX, router);
op_ima_obj_attr_get (router_obj, "MPLS Parameters", &mps_params_comp_objid);
mpls_params_row_objid = op_topo_child (mps_params_comp_objid , OPC_OBJTYPE_GENERIC
, 0);
op_ima_obj_attr_get (mpls_params_row_objid, "Traffic Mapping Configuration", &
trunk_comp_objid);

trunk_row_objid=op_topo_child (trunk_comp_objid , OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (trunk_row_objid , "LSP",&trunk_comp_objid);

trunk_row_objid=op_topo_child (trunk_comp_objid , OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (trunk_row_objid , "Primary LSPs",&trunk_comp_objid);

trunk_row_objid=op_topo_child (trunk_comp_objid , OPC_OBJTYPE_GENERIC, 0);
op_ima_obj_attr_get (trunk_row_objid , "LSP Name" ,nomLSP);

```

```

sprintf(temporalSite, "%s = %d //" , nomLSP, num_sites);
fprintf(fSite, "%s \n" , temporalSite);
fclose(fSite);

/*fSite = archivo para los nombres de los sitios*/
fSite=fopen(" ruta/nomSitios.txt" ,"ab");

for(i=0;i<num_sites;i++)
{
    data[i] = (SitesData*) op_prg_mem_alloc (sizeof (SitesData));
    sites_row_objid = op_topo_child (sites_comp_objid , OPC_OBJTYPE_GENERIC, i);
    op_ima_obj_attr_get (sites_row_objid , "Site name" , &data[i]->site_name);
    op_ima_obj_attr_get (sites_row_objid , "Interface In" , &data[i]->interface_in);
    data[i]->site_obj = op_id_from_name(0, OPC_OBJTYPE_NODE_FIX, data[i]->site_name
    );
    data[i]->appl_obj = op_id_from_name(data[i]->site_obj , OPC_OBJTYPE_PROC, "
    application");
    op_ima_obj_attr_get (data[i]->site_obj , "IP Host Parameters" , &
    mps_params_comp_objid);
    mpls_params_row_objid = op_topo_child (mps_params_comp_objid ,
    OPC_OBJTYPE_GENERIC, 0);
    op_ima_obj_attr_get (mpls_params_row_objid , "Interface Information" , &
    trunk_comp_objid);
    trunk_row_objid = op_topo_child (trunk_comp_objid , OPC_OBJTYPE_GENERIC, 0);
    op_ima_obj_attr_get (trunk_row_objid , "Address" , &data[i]->src_addr);

    /*Guardando el nombre de los sitios en un archivo*/
    op_ima_obj_attr_get(op_id_self(), "Router" , router);

    router_obj = op_id_from_name(0, OPC_OBJTYPE_NODE_FIX, router);
    op_ima_obj_attr_get (router_obj , "MPLS Parameters" , &mps_params_comp_objid);
    mpls_params_row_objid = op_topo_child (mps_params_comp_objid ,
    OPC_OBJTYPE_GENERIC, 0);

```

```

op_ima_obj_attr_get ( mpls_params_row_objid , " Traffic Mapping Configuration" , &
    trunk_comp_objid );

trunk_row_objid=op_topo_child ( trunk_comp_objid , OPC_OBJTYPE_GENERIC , 0 );
op_ima_obj_attr_get ( trunk_row_objid , "LSP" ,&trunk_comp_objid );

trunk_row_objid=op_topo_child ( trunk_comp_objid , OPC_OBJTYPE_GENERIC , 0 );
op_ima_obj_attr_get ( trunk_row_objid , " Primary LSPs" ,&trunk_comp_objid );

trunk_row_objid=op_topo_child ( trunk_comp_objid , OPC_OBJTYPE_GENERIC , 0 );
op_ima_obj_attr_get ( trunk_row_objid , "LSP Name" ,nomLSP );

//nomLSP = indica a que traffic trunk pertenece el sitio
sprintf(temporalSite," %s = %s //" ,nomLSP,data[i]->site_name); //Guarda el lsp y
    nombre del sitio
fprintf(fSite," %s \n" ,temporalSite);

}
fclose(fSite);

if ( op_ima_obj_attr_get( op_id_self(), " Router", router) == OPC_COMPCODE_FAILURE)
    change_attr_error("Get Router name attribute failed");

op_sim_message(" Leido el Router: ", router);
router_obj = op_id_from_name(0, OPC_OBJTYPE_NODE_FIX, router);
router_processor = op_id_from_name(router_obj, OPC_OBJTYPE_PROC, "ip");
sprintf (temporal1, "El id del router es: %d", router_obj);
if (router_obj == OPC_OBJID_INVALID )
    change_attr_error("Error in retriving ID's");
else op_sim_message("ID's ok", temporal1);

if ( op_ima_obj_attr_get( op_id_self(), " Read_Policies_Interval", &read_intvl) ==
    OPC_COMPCODE_FAILURE)

```

```

change_attr_error("Get read_interval attribute failed");

if (op_ima_obj_attr_get(op_id_self(), "Policy_Filename", temporal1) ==
    OPC_COMPCODE_FAILURE)
    change_attr_error("Get Policy_Filename attribute failed");
else sprintf (policy_filename, "ruta/%s", temporal1);

op_ima_obj_attr_get (router_obj, "MPLS Parameters", &mps_params_comp_objid);
mpls_params_row_objid = op_topo_child (mps_params_comp_objid, OPC_OBJTYPE_GENERIC
, 0);
op_ima_obj_attr_get (mpls_params_row_objid, "Traffic Mapping Configuration", &
    trunk_comp_objid);
num_tt = op_topo_child_count (trunk_comp_objid, OPC_OBJTYPE_GENERIC);

op_intrpt_schedule_self(op_sim_time() + 30, READ_POLICIES_CODE);

```

En este proceso también se modificó el estado *change*, dentro del *Enter Executives* de este estado se modificó el siguiente código:

```

f_SLSI_Policy = fopen(policy_filename, "r+");
f=0;
while ((fscanf(f_SLSI_Policy, "%[^\n]", read_SLSI_Policy)) != EOF)
{
    fgetc(f_SLSI_Policy); //Reads in '\n' character and moves file stream past
    delimiting character
    f++;
    temporal1 = "TRUE";
    c[0] = *temporal1;
    temporal2 = strstr(read_SLSI_Policy, temporal1);
    if(temporal2!=NULL)
    {
        temporal3 = strstr(read_SLSI_Policy, router);
        if(temporal3!=NULL)

```

```

    {
temporal1 = "Flow";
temporal2 = strstr(read_SLSI_Policy , temporal1);
temporal1 = "=";
temporal4 = strstr(temporal2 , temporal1);
temporal4 = temporal4+2;
for(i=0;strcmp(c,"-")!=0;i++)
{
    temporal[i] = *temporal4;
    temporal4 = temporal4+1;
    c[0] = *temporal4;
}
temporal[i]='\0';
for(i=0;i<num_sites;i++)
{
    if(strcmp(temporal , data[i]->src_addr)==0)
    {
        sprintf (interface , "%d" , data[i]->interface_in);
        site_index = i;
        op_sim_message("Entre en interface", "");
    }
}

temporal1 = "STOP";
temporal2 = strstr(read_SLSI_Policy , temporal1);
if(temporal2!=NULL)
{
    data[site_index]->stop = 1;
}
else{
    data[site_index]->stop = 0;
}
temporal1 = "New";

```

```

temporal2 = strstr(read_SLSI_Policy, temporal1);
temporal1 = "=";
temporal4 = strstr(temporal2, temporal1);
temporal4 = temporal4+2;
for(i=0;strcmp(c,"s")!=0;i++)
{
    data[site_index]->new_tt[i] = *temporal4;
    c[0] = *temporal4;
    temporal4 = temporal4+1;
}
data[site_index]->enforcement = 1;
data[site_index]->line = f;
    }
}
}

fclose (f_SLSI_Policy);
op_ima_obj_attr_get (router_obj, "MPLS Parameters", &mps_params_comp_objid);
mpls_params_row_objid = op_topo_child (mps_params_comp_objid, OPC_OBJTYPE_GENERIC
, 0);
op_ima_obj_attr_get (mpls_params_row_objid, "Traffic Mapping Configuration", &
trunk_comp_objid);
for(i=0;i<num_sites;i++)
{
    if(data[i]->enforcement == 1)
    {
        op_sim_message(" Actuamos sobre ", data[i]->site_name);
        invoke = 1;
        temporal1 = "";
        temporal2 = " Policy Enforcement = FALSE ";
        temporal3 = op_prg_mem_alloc(500);
        f_SLSI_Policy = fopen(policy_filename, "r+");
        linea = data[i]->line;

```

```

for (j=0;j<linea;j++)
{
    fscanf(f_SLSI_Policy , "%[^\n]" , read_SLSI_Policy);
    fgetc(f_SLSI_Policy);
}
fclose (f_SLSI_Policy);
temporal1=strstr(read_SLSI_Policy , "//");
sprintf (temporal3 , "%6%6\n" ,temporal2 ,temporal1);
f_SLSI_Policy = fopen(policy_filename , "r+");
for (j=0;j<linea -1;j++)
{
    fscanf(f_SLSI_Policy , "%[^\n]" , read_SLSI_Policy);
    fgetc(f_SLSI_Policy);
}

fprintf(f_SLSI_Policy ,temporal3);
fclose (f_SLSI_Policy);
sprintf (interface , "%d" , data[i]->interface_in);
op_sim_message("Cambiamos a:" , data[i]->new_tt);

for (j=0;j<num_tt;j++)
{
    trunk_row_objid = op_topo_child (trunk_comp_objid , OPC_OBJTYPE_GENERIC , j)
    temporal1 = op_prg_mem_alloc(100);
    op_ima_obj_attr_get (trunk_row_objid , "Interface In" , temporal1);
    if (strstr(temporal1 , interface)!=NULL)
    {
        op_ima_obj_attr_set (trunk_row_objid , "Traffic Trunk" , data[i]->new_tt);
    }
}

if (data[i]->stop==1)
{

```

```

        op_intrpt_schedule_remote(op_sim_time() + 1, Stop_Service_Code , data[i]->
            appl_obj);
    }
    else
        op_intrpt_schedule_remote(op_sim_time() + 1, Start_Service_Code , data[i]->
            appl_obj);
    data[i]->enforcement = 0;
}
}

if (invoke==1)
{
    op_sim_message(" Reiniciamos ", "");
    op_intrpt_schedule_remote(op_sim_time() + 1, REINIT_CODE, router_processor);
    invoke = 0;
}

op_intrpt_schedule_self(op_sim_time() + read_intvl , READ_POLICIES_CODE);

```

B.1.3 Proceso ip_dispatch

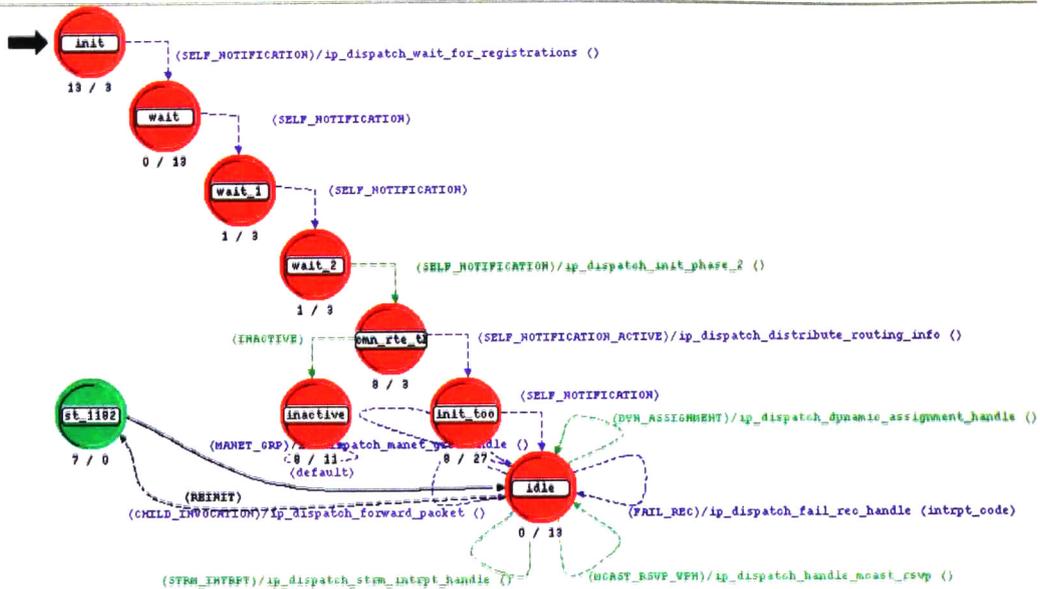
Los cambios de configuración que nos interesa realizar en los routers, actúan sobre el módulo de IP de los nodos LER de la red. Este módulo IP se rige por el proceso *ip_dispatch* (ver figura B.3).

Dentro del estado *st_1182* se agregó lo siguiente:

```

ip_dispatch_mpls_info_read ();
mpls_support_iface_mpls_info_set (module_data.node_id , module_data .
    interface_table_ptr , &module_data);
ip_dispatch_interface_mpls_init ();
module_data mpls_info_ptr->mgr_prohandle = op_pro_create(" mpls_mgr",&module_data)
;
op_pro_invoke (module_data.mpls_info_ptr->mgr_prohandle , OPC_NIL);
ip_support_module_data_htable_entry_add (&module_data);

```

Figura B.3: Proceso *ip_dispatch*

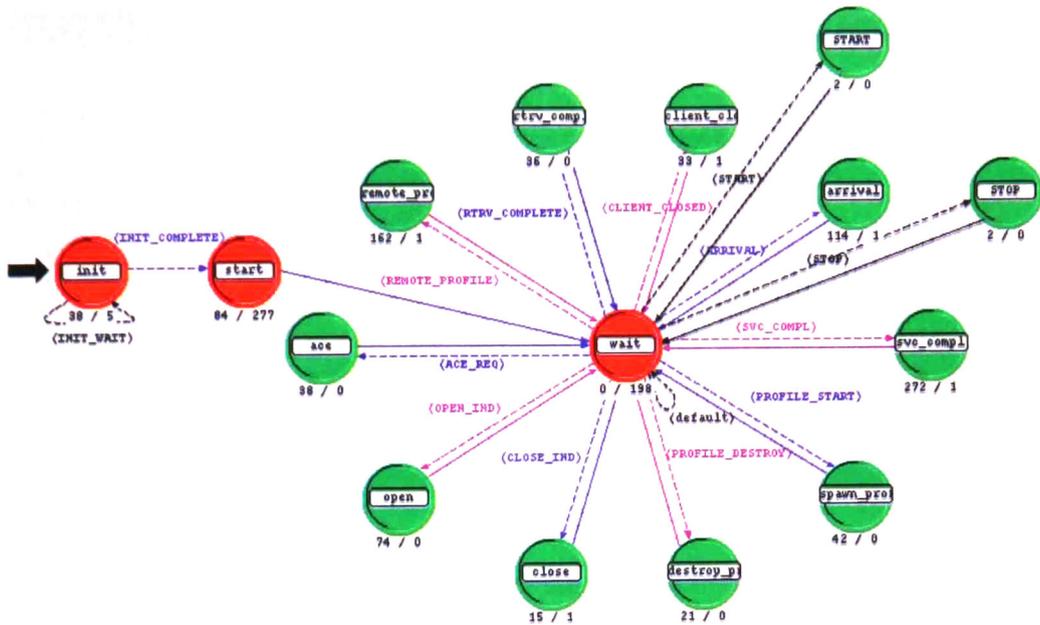
```
op_sim_message(" Reinitilise Module Date" , "");
```

B.1.4 Proceso *gna_clsvr*

El proceso *gna_clsvr* mostrado en la figura B.4 fue modificado para obtener el número de usuarios activos, aceptados y rechazados que pertenecen a cierto perfil.

Dentro del estado *start*, después de la función *gna_clsvr_mgr_sv_init()* se agregó el siguiente código:

```
stop=0;
active_profiles=0;
discarted_profiles=0;
accepted_profiles=0;
temporal1 = op_prg_mem_alloc(100);
site_name = op_prg_mem_alloc(500);
```

Figura B.4: Diagrama de estados del proceso *gna_clsvr*

```

op_ima_obj_attr_get(my_node_id, "name", temporal1);
sprintf (site_name, "ruta/%s.txt", temporal1);

if (op_ima_obj_attr_exists (my_objid, "Aceptacion"))
    op_ima_obj_attr_get(my_objid, "Aceptacion", &aceptacion);
else aceptacion=100;
rnd_gen = op_prg_random_gen_create (descartado);

```

En el estado *destroy_profile* se insertó al final el siguiente código:

```

active_profiles --;
f1 = fopen(site_name, "ab");
temporal1 = op_prg_mem_alloc(100);
sprintf(temporal1,"activos %d aceptados %d rechazados %d time %f",
    active_profiles, accepted_profiles, discarded_profiles, op_sim_time());
fprintf(f1,"%e \n",temporal1);
fclose(f1);

```

Dentro del estado *spawn_profile* se modificó lo siguiente:

```

rand_int = (op_prg_random_integer_gen (rnd_gen) %100) + 1;
if(rand_int<=aceptacion)
    descartado=0;
else descartado=1;

if(stop!=1&&descartado==0)
{
    gna_profile_spawn (intrpt_code);
    active_profiles++;
    accepted_profiles++;
}
else discarded_profiles++;

f1 = fopen(site_name , "ab");
temporal1 = op_prg_mem_alloc(100);
sprintf(temporal1," activos %d aceptados %d rechazados %d time %f",active_profiles
    , accepted_profiles , discarded_profiles , op_sim_time());
fprintf(f1," %e \n" ,temporal1);
fclose(f1);

```

En el estado *START* se agregó el siguiente código:

```

op_sim_message(" desactivo stop-", "");
stop = 0;

```

Por último, dentro del estado *STOP* se modificó lo siguiente:

```

stop = 1;
op_sim_message(" activo stop-", "");

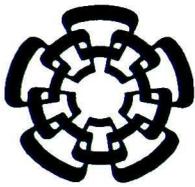
```

Bibliografía

- [1] <http://www.opnet.com/>.
- [2] A.Manzalini, P.H. Deussen, S. Nechifor, M. Mamei, R. Minerva, C. Moiso, A. Salden, T. Wauters, and F. Zambonelli. "Self-optimized Cognitive Network of Networks". *Computer Journal*, February 2010.
- [3] Carlos A. Coello Coello. "An Updated Survey of GA-Based Multiobjective Optimization Techniques". *ACM Computing Surveys*, Vol. 32, June 2000.
- [4] Mario Garza Fabre. "Many-objective optimization through evolutionary algorithms" Master's thesis, Laboratorio de Tecnologías de la Información, CINVESTAV, Tamaulipas, México, September 2009.
- [5] Gerardo Montemayor García. "On the Use of Surrogated Models in Multiobjective Evolutionary Algorithms". Master's thesis, Laboratorio de Tecnologías de Información, CINVESTAV, Tamaulipas, México, December 2011.
- [6] D.R. Jones, M. Schonlau, and W.J. Welch. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global optimization*, vol. 13(No. 4):pp. 455–492, 1998.
- [7] J.Rubio-Loyola, J. Serrat, M. Charalambides, P.Flegkas, and G. Pavlou. "A Methodological Approach towards the Refinement Problem in Policy-based Management Systems" *IEEE Communications Magazine*, Vol. 44(No. 10):pp. 60–68, October 2006.
- [8] Joshua Knowles. "ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems". *IEEE Transactions on Evolutionary Computation*, Vol. 10(No. 1):pp. 50–66, January 2006.

- [9] Alan Díaz Manríquez. "Optimization of dynamic multiobjective problems using evolutionary algorithms". Master's thesis, Laboratorio de Tecnologías de la Información, CINVESTAV, Tamaulipas, México, November 2009.
- [10] Efren Mezura Montes and Carlos A. Coello Coello. "Conceptos de Optimización Multiobjetivo para el Manejo de Restricciones en Algoritmos Evolutivos: Un estudio Comparativo". *Proceedings of the 1st Mexican Conference on Evolutionary Computation (COMCEV 2003)*, pages 1–12, May 2003.
- [11] Eleni Mykoniati, Charalampos Charalampous, Panos Georgatsos, Takis Damilatis, Danny Goderis, Panos Trimintzios, George Pavlou, and David Griffin. "Admission Control for Providing QoS in DiffServ IP Networks: The TEQUILA Approach". *IEEE Communications Magazine*, January 2003.
- [12] Javier Rubio Loyola, Marinos Charalambides, Issam Aib, Joan Serrat, George Pavlou, and Raouf Boutaba. "Business-driven Management of Differentiated Services". *Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium IEEE/IFIP NOMS 2010*, April 2010.
- [13] P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, D. Griffin, P. Georgatsos, D. Goderis, Y. T'Joens, L. Georgiadis, C. Jacquenet, and R. Egan. "A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks". *IEEE Communications Magazine*, May 2001.
- [14] Panos Trimintzios, George Pavlou, Paris Flegkas, Panos Georgatsos, Abolghasem (Hamid) Asgari, and Eleni Mykoniati. "Service-Driven Traffic Engineering for Intradomain Quality of Service Management" *IEEE Network*, June 2003.
- [15] Kostas Tsagkaris, Panagiotis Demestichas, and George Dimitrakopoulos. "Policies for the Management of Services in CDMA-Based Segments of the B3G World". *IEEE Vehicular Technology Magazine*, September 2007.

-
- [16] Eckart Zitzler and Lothar Thiele. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach". *Evolutionary Computation*, Vol. 3(No. 4):pp. 257–271, November 1999.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL IPN

UNIDAD TAMAULIPAS

Cd. Victoria, Tamaulipas, a 24 de febrero de 2012

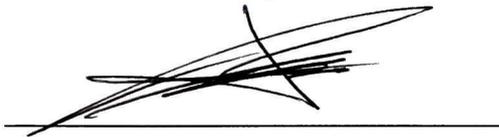
Los abajo firmantes, integrantes del jurado para el examen de grado que sustentará el C. ENDA MARIEL MIL CHONTAL, declaramos que hemos revisado la tesis titulada:

“Optimización de indicadores de negocio para servicios de próxima generación de redes cognitivas”

Y consideramos que cumple con los requisitos para obtener el grado de Maestra en Ciencias en Computación.

Atentamente,

Dr. Javier Rubio Loyola



Dr. Víctor Jesús Sosa Sosa



Dr. Gregorio Toscano Pulido





CINVESTAV - IPN
Biblioteca Central



SSIT0010850