

Centro de Investigación Científica y de Educación Superior de Ensenada



**ESTUDIO DE MECANISMOS DE CONTROL DE
FLUJO PARA LA ENTREGA EFICIENTE Y
ROBUSTA DE SERVICIOS EN LA RED
INTELIGENTE (RI)**

**TESIS
MAESTRIA EN CIENCIAS**

JESUS MARTIN BEDOY SOLIS

ENSENADA, BAJA CALIFORNIA, MEXICO.

SEPTIEMBRE DE 1997.

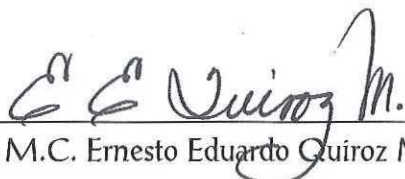
BIBLIOTECA
CICESE

19526

TK
5105
.5
.B43
G.2

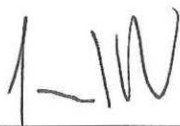
TESIS DEFENDIDA POR
JESÚS MARTÍN BEDOY SOLÍS

Y APROBADA POR EL SIGUIENTE COMITE



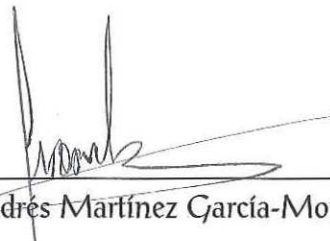
M.C. Ernesto Eduardo Quiroz Morones

Director del Comité



Dr. Jesús Favela Vara

Miembro del Comité



Dr. Ciro Andrés Martínez García-Moreno

Miembro del Comité



Dr. José Luis Medina Monroy

*Jefe del Departamento de Electrónica y
Telecomunicaciones*



Dr. Federico Graef Ziehl

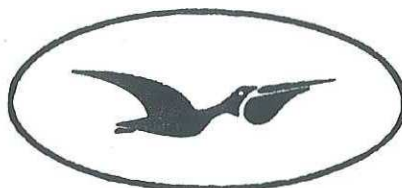
Director de Estudios de Posgrado

5 de septiembre de 1997

19526

BIBLIOTECA
CICESE

**CENTRO DE INVESTIGACIÓN CIENTÍFICA Y
DE EDUCACIÓN SUPERIOR DE ENSENADA**



CICESE

DIVISIÓN DE FÍSICA APLICADA

**DEPARTAMENTO DE
ELECTRÓNICA Y TELECOMUNICACIONES**

**ESTUDIO DE MECANISMOS DE CONTROL DE
FLUJO PARA LA ENTREGA EFICIENTE Y
ROBUSTA DE SERVICIOS EN LA RED
INTELIGENTE (RI)**

TESIS

**Que para cubrir parcialmente los requisitos necesarios para
obtener el grado de Maestro en Ciencias presenta**

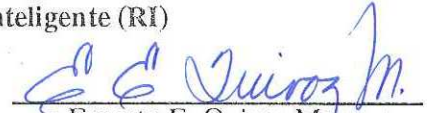
JESÚS MARTÍN BEDOY SOLIS

Ensenada, B.C. México, septiembre de 1997

Resumen de tesis de Jesús Martín Bedoy Solís, presentada como requisito parcial para la obtención del grado de MAESTRO en CIENCIAS con especialidad en ELECTRÓNICA Y TELECOMUNICACIONES. Ensenada, Baja California, México, agosto de 1997.

**Estudio de mecanismos de control de flujo para la entrega
eficiente y robusta de servicios en la Red Inteligente (RI)**

Resumen aprobado por:


Ernesto E. Quiroz Morones
Director de Tesis

La Red Inteligente (RI) es un concepto de arquitectura que nos permite de una manera flexible, rápida y económica el diseño y desarrollo de nuevos servicios en redes de telecomunicaciones existentes y futuras. Con el fin de disponer de estos servicios con una mínima modificación en la red, la introducción de nuevos servicios en la RI deberá de ser transparente al usuario y proveedor de la red. En la RI, la introducción de nuevos servicios dejan de estar solamente en el hardware y software disponible en el conmutador de control por programa almacenado. Ahora, mucha de la inteligencia esta centralizada en bases de datos las cuales son accesadas a través de redes de señalización de conmutación de paquetes. Con este concepto, el control de la información de la llamada se lleva a cabo entre el punto de conmutación de servicios (SSP, Service Switched Point) y las bases de datos localizadas en el punto de control de servicio (SCP, Service Control Point).

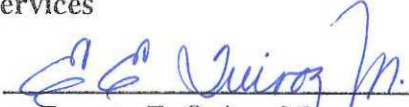
El SCP es el núcleo de la arquitectura de RI. Este sirve como un servidor lógico en la arquitectura de RI conteniendo la mayoría de la información y la lógica de servicio. En la RI, el SCP y el SSP coordinan su operación a través del intercambio de mensajes vía la red del sistema de señalización por canal común número 7 (CCSS7, Common Channel Signaling System number 7). Para algunos servicios el SCP puede requerir el uso de periféricos inteligentes (IP, Intelligent Peripherals) o de bases de datos remotas. En un futuro muy cercano, la introducción de nuevos y mas avanzados servicios de RI implicará no solo un incremento en el tráfico de señalización, sino también un incremento en la carga ofrecida hacia el SCP. Por lo tanto, la centralización de los programas de lógica de servicio para llevar a cabo el proceso de llamadas en la RI y otras funciones llevadas a cabo en el SCP pueden generar un retardo adicional e inaceptable en el tiempo de establecimiento de una llamada y el decremento en el caudal eficaz del sistema.

En este trabajo de tesis, se desarrollan modelos de colas para representar la arquitectura de RI y evaluar el desempeño de dos esquemas de control de flujo impuesto al SCP, llamados, Control por Bandas de Llamadas (Call Gapping) y Control de Ventana (Window), asimismo se introduce un mecanismo de control de congestión a dos niveles para garantizar una mejor calidad de servicio (QoS, Quality of Service) de RI. Nuestro trabajo se diferencia a estudios previos, en que consideramos en detalle la arquitectura interna del SCP consistiendo de múltiples procesadores enlazados por una red de alta velocidad y compartiendo una base de datos común. Además, consideramos la operación del sistema implementando servicios de RI a varios niveles de complejidad. Ésta se basa en el número de visitas requeridas a la base de datos para completar el servicio. Estamos particularmente interesados en evaluar el tiempo de respuesta de una llamada y el caudal eficaz en el sistema, siendo estos dos parámetros los de mayor interés para el usuario y el proveedor, respectivamente.

ABSTRACT of thesis by Jesús Martín Bedoy Solis, presented as partial requirement to obtain the MASTER OF SCIENCE degree with specialization in Electronics and Telecommunications. Ensenada, Baja California, Mexico, August 1997.

**Study of load control mechanisms for the
effective and robust delivery of IN services**

Abstract aproved by:


Ernesto E. Quiroz Morones
Thesis Director

The Intelligent Network (IN) is an architectural concept enabling the flexible, timely, and cost effective design and deployment of novel services in existing and future networks. With the aim of enabling the provision of novel services with the minimum modification to the network, ideally the introduction of new facilities should be 'transparent' to the user and network provider. In IN, the introduction of new facilities cease to be solely dependent upon the hardware and software in stored program control switches. Rather, much of the intelligence is centralized in databases which are accessed through packet switched signaling networks. With this concept, call control information is deployed between a Service Switched Point (SSP) and databases located in a Service Control Point (SCP).

The SCP is an important resource in the IN architecture. It serves as a logical hub in the IN architecture containing most of the service logic and associated data. Under IN, the SCP and SSP coordinate their operation through message exchange via the Common Channel Signaling System number 7 (CCSS7). For some services the SCP may require the use of Intelligent Peripherals (IP) or remote Databases. In the near future, the introduction of new and more advanced IN services will imply not only an increase on the signaling traffic, but an increase on the load submitted to the SCP. Therefore, the centralization of service logic programs used to provide IN call processing and other house-keeping functions performed at the SCP may create bottleneck problems.

In this thesis, we develop queuing models for the SCP and evaluate the performance of two schemes to control the load imposed to the SCP, namely, Call Gapping and Window Control and introduce a two level congestion control mechanism to provide better quality of services guarantees to IN services. Our study differs from previous studies in that we consider in detail an internal SCP architecture consisting of multiple processors linked by a high-speed network and sharing a common database. Furthermore, we consider the operation of the system when servicing IN services of various levels of complexity. The complexity being measured by the number of visits to the database required to complete the service. We are particularly interested in evaluating the response time to a call and system throughput since these are the two performance metrics of concern to the users and service provider, respectively.

DEDICATORIA

A DIOS, que sin su ayuda nada hubiera sido posible.

A las personas más importantes en mi vida, mis padres, José Luis Bedoy Velazco y Ramona Solis de Bedoy; gracias por darme todas las oportunidades para seguir adelante, en especial por darme la vida; los quiero.

A mi familia, por darme el apoyo y las fuerzas necesarias para concluir mis estudios, por ser más que nada mis amigos, en especial a mi hermano "Pepe", a mis sobrinos y sobrinas y a mi pequeña "Michelle" los cuales han servido como centro de inspiración en el transcurso de mi formación profesional.

A mis mejores amigas, Paty, Sonia y Lulú, por sus consejos, atenciones y por haberme enviado tantas tarjetas en el tiempo que estuve lejos, las cuales me llenan de energía y fuerzas para seguir adelante.

AGRADECIMIENTOS

Gracias....

Al Doctor Luis Orozco Barbosa, por haberme brindado la oportunidad de trabajar bajo su supervisión en la Universidad de Ottawa, Canadá, por sus consejos, paciencia y atenciones a lo largo de este trabajo de tesis.

A la Universidad de Ottawa, principalmente a todo el equipo de trabajo del MCRLab, en especial al Doctor Nicolas D. Georganas, por haberme abierto las puertas y darme la libertad de hacer uso de sus instalaciones y equipo para llevar a cabo mi investigación.

A Michelle Ouellette y al "Gachupin" Pedro Cuencas, por su amistad y ayuda a lo largo de esta tesis. A Barathi, por su amistad y por todos sus comentarios y consejos, gracias por ayudarme a mejorar mi inglés.

Al "Pequeño", Miguel López por su ayuda, atención, dedicación, comentarios y por haber revisado la escritura de la tesis.

A todos los Telecos; Héctor, Teacher, Rivera, Pancho, Tamayo, Luis, en especial a Edith, por su amistad y consejos. Sin olvidar también a todos los demás compañeros de generación de Control y Altas Frecuencias, en especial a Carmelita, y a los ya no tan nuevos Telecos en especial a Guille. Gracias por ser mis amigos.

A todos mis amigos y ex-compañeros de Mexicali y Ensenada, en especial a Silvita, Any, Edoardo, Chencho, Carrillo, Otton, Elba, Edith, Claudia, Benny, Claudita, Eliot, Ely, Maricela de la O, Gerardo, Rubén, Cinthia, Carlos y a todos aquellos que me han brindado su amistad lo cual me ha permitido seguir adelante con muchas ganas.

A todos mis profesores, porque gracias a ellos he podido culminar con esta etapa más de mi vida, en especial al M. C. Ernesto Quiroz Morones por ser mi director de tesis y por la confianza que tuvo en mí para poder realizar este trabajo. Al Doctor Favela y al Doctor Ciro por formar parte de mi comité de tesis y por la ayuda brindada en la revisión del manuscrito.

Al departamento de Servicios Escolares en especial a Margarita Jauregui y a Citlalí, por su valiosa ayuda y apoyo en el transcurso de la maestría.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al CICESE, mis más sinceros agradecimientos por todo el apoyo brindado para la realización de mis estudios de maestría.

CONTENIDO (Continuación)

Página

II.4.4 PARTE DE APLICACIÓN DE RED INTELIGENTE (INAP, IN Application Part).....	37
II.5 CONJUNTO DE CAPACIDADES DE LA ITU-T PARA RED INTELIGENTE ...	39
III. ESTUDIO DE MECANISMOS DE CONTROL	45
III.1 INTRODUCCIÓN.....	45
III.2 DEFINICIÓN DE MECANISMO DE CONTROL DE FLUJO.....	46
III.3 DEFINICIÓN DE MECANISMO DE CONTROL DE CONGESTIÓN.....	48
III.4 DIFERENCIA ENTRE CONTROL DE FLUJO Y CONTROL DE CONGESTIÓN.	50
III.5 DESCRIPCIÓN DE MECANISMOS DE CONTROL EN LA RED INTELIGENTE.....	51
III.5.1 CONTROL DE BANDAS DE LLAMADAS (CALL GAPPING).....	51
III.5.2 CONTROL DE LA VENTANA (WINDOW)	55
III.5.3 CONTROL A DOS NIVELES (TWO LEVELS CONTROL).....	58
IV. MODELADO Y SIMULACIÓN DE MECANISMOS DE CONTROL EN LA RED INTELIGENTE	61
IV.1 INTRODUCCIÓN.....	61
V.2 DESCRIPCIÓN DEL MODELO BAJO ESTUDIO	62
IV.2.1 DESCRIPCIÓN GENERAL.....	62
IV.2.2 CON CONTROL DE BANDA DE LLAMADAS (CALL GAPPING)	69
IV.2.3 CON CONTROL DE LA VENTANA (WINDOW).....	77
IV.2.4 CON CONTROL A DOS NIVELES (TWO LEVELS CONTROL)	82
V. ANÁLISIS DE RESULTADOS.....	86
V.1. INTRODUCCIÓN.....	86

CONTENIDO(Continuación)

	Página
V.2 MODELO DE TRÁFICO	87
V.3 FASE I : SIN CONTROL	91
V.3.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP	91
V.3.2 RETARDO DE LLAMADAS INTELIGENTES	95
V.3.3 GOODPUT EN EL SCP	101
V.4 FASE I : CON CONTROL DE LA VENTANA (WINDOW)	103
V.4.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP	103
V.4.2 RETARDO DE LLAMADAS INTELIGENTES	108
V.4.3 GOODPUT EN EL SCP	113
V.5 FASE I : CON CALL GAPPING	115
V.5.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP	115
V.5.2 RETARDO DE LLAMADAS INTELIGENTES	120
V.5.3 GOODPUT EN EL SCP	124
V.6 FASE II: MECANISMO DE CONTROL DE CONGESTIÓN A DOS NIVELES.	126
V.6.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP	126
V.6.2 RETARDO DE LLAMADAS INTELIGENTES	131
V.6.3 GOODPUT EN EL SCP	136
V.7 DIFERENCIAS ENTRE WINDOW, CALL GAPPING Y DOS NIVELES.	136
VI. CONCLUSIONES, RECOMENDACIONES Y COMENTARIOS	142
VI.1 CONCLUSIONES	142
VI.2 RECOMENDACIONES	144
VI.3 COMENTARIOS	145
LITERATURA CITADA	147
ANEXO A	151

CONTENIDO(Continuación)

Página

DESCRIPCIÓN GENERAL DEL MÉTODO DE SIMULACIÓN OPNET	151
GLOSARIO	154



LISTA DE FIGURAS

Figura	<u>Página</u>
1.- ARQUITECTURA DE LA RIA.....	5
2.- ARQUITECTURA MULTIPROCESADORES PARA EL SCP PROPUESTA.	7
3.- MODELO CONCEPTUAL DE RED INTELIGENTE (ITU-T REC. Q.1201, 1992).....	13
4.- PLANO FUNCIONAL DISTRIBUIDO.	16
5.- ARQUITECTURA FÍSICA DE LA RED INTELIGENTE BASADA EN EL SISTEMA DE SEÑALIZACIÓN NÚMERO 7 (SS7)	20
6.- COMPONENTES DEL SSP.....	23
7.- COMPONENTES DEL SCP.	26
8.- COMPARACIÓN DEL MODELO DE REFERENCIA OSI CON EL PROTOCOLO SS7.	31
9.- RELACIÓN DEL PROTOCOLO SS7 EN LAS ENTIDADES FÍSICAS DE RED INTELIGENTE	32
10.- ESTRUCTURA GENERAL DEL MTP EN LA RED SS7.....	34
11.- ESTRUCTURA GENERAL DEL SCCP EN LA RED SS7.	35
12.- ESTRUCTURA DE TCAP - ASE.	36
13.- ARQUITECTURA DEL PROTOCOLO INAP.....	38
14.- ITU-T RI-CS ESTANDARIZACIÓN.	40
15.- ITU-T ESTRUCTURA DE RECOMENDACIONES PARA LA RI.	41
16.- DESEMPEÑO DEL SISTEMA Vs CARGA OFRECIDA.....	45
17.- MECANISMO DE CONTROL CALL GAPPING.....	53
18.- MECANISMO DE CONTROL DE LA VENTANA.....	56
19.- DIAGRAMA DE TIEMPO DEL MECANISMO DE CONTROL A DOS NIVELES.	60
20.- ARQUITECTURA DE RI BAJO ESTUDIO A NIVEL NODO.	62
21.- MODELOS DE COLAS DE LA RED INTELIGENTE.	63
22.- CONFIGURACIÓN A BLOQUES DE RI CON CONTROL DE CALL GAPPING.....	70
23.- PROBABILIDAD DE BLOQUEO EN EL SCP.....	73
24.- PROBABILIDAD DE BLOQUEO EN EL SSP.	74
25.- REACCIÓN DEL MECANISMO DE CONTROL CALL GAPPING EN EL SSP	76

LISTA DE FIGURAS(Continuación)

Figura	<u>Página</u>
26.- CONFIGURACIÓN A BLOQUES DE RI CON CONTROL DE LA VENTANA.	77
27.- COMPORTAMIENTO DEL MECANISMO DE CONTROL DE LA VENTANA EN EL SSP.	80
28.- CONFIGURACIÓN A BLOQUES DE RI CON CONTROL DE CONGESTIÓN A DOS NIVELES.	83
29.- TRÁFICO MONITOREADO EN LA RED INTELIGENTE EN STENTOR RESOURCE CENTER.	88
30.- TRÁFICO BAJO ESTUDIO.	90
31.- NÚMERO DE LLAMADAS EN EL CP CON RESPECTO AL TIEMPO DE SIMULACIÓN.	92
32.- NÚMERO DE LLAMADAS INTELIGENTES EN ESPERA DE SER ATENDIDAS POR EL CPU.	93
33.- NÚMERO DE LLAMADAS INTELIGENTES EN ESPERA DE SER ATENDIDAS POR LDB.	93
34.- UTILIZACIÓN DEL CPU.	94
35.- UTILIZACIÓN DE LDB.	94
36.- RETARDO DE LLAMADAS INTELIGENTES.	95
37.- DISTRIBUCIÓN DE RETARDO DE LLAMADAS INTELIGENTES.	101
38.- GOODPUT EN EL CP.	103
39.- NÚMERO DE LLAMADAS EN EL CP CON RESPECTO AL TIEMPO DE SIMULACIÓN UTILIZANDO EL CONTROL DE LA VENTANA.	104
40.- NÚMERO DE LLAMADAS INTELIGENTES EN ESPERA DE SER ATENDIDAS POR EL CPU UTILIZANDO EL MECANISMO DE LA VENTANA.	105
41.- NÚMERO DE LLAMADAS INTELIGENTES EN ESPERA DE SER ATENDIDAS POR LDB UTILIZANDO EL MECANISMO DE LA VENTANA.	106
42.- UTILIZACIÓN DEL CPU.	107
43.- UTILIZACIÓN DE LDB.	107
44.- RETARDO DE LLAMADAS INTELIGENTES CON CONTROL DE LA VENTANA.	108
45.- DISTRIBUCIÓN DE RETARDO DE LLAMADAS INTELIGENTES CON CONTROL DE LA VENTANA.	113
46.- GOODPUT EN EL CP CON CONTROL DE LA VENTANA.	114
47.- NÚMERO DE LLAMADAS EN EL CP CON RESPECTO AL TIEMPO DE SIMULACIÓN UTILIZANDO EL MECANISMO DE CONTROL CALL GAPPING.	116

LISTA DE FIGURAS(Continuación)

Figura	<u>Página</u>
48.- <i>NÚMERO DE LLAMADAS EN ESPERA DE SER ATENDIDAS POR EL CPU CON CALL GAPPING.</i>	117
49.- <i>NÚMERO DE LLAMADAS EN ESPERA DE SER ATENDIDAS POR LDB CON CALL GAPPING.....</i>	118
50.- <i>UTILIZACIÓN DEL CPU.....</i>	118
51.- <i>UTILIZACIÓN DE LDB.....</i>	118
52.- <i>RETARDO DE LLAMADAS INTELIGENTES UTILIZANDO EL MECANISMO DE CONTROL CALL GAPPING.....</i>	120
53.- <i>DISTRIBUCIÓN DE RETARDO DE LLAMADAS INTELIGENTES CON CALL GAPPING.....</i>	124
54.- <i>GOODPUT EN EL CP CON CALL GAPPING.....</i>	125
55.- <i>NÚMERO DE LLAMADAS EN ESPERA DE SER ATENDIDAS POR EL CPU CON CONTROL DE CONGESTIÓN A DOS NIVELES.....</i>	127
56.- <i>NÚMERO DE LLAMADAS EN ESPERA DE SER ATENDIDAS POR LDB CON CONTROL A DOS NIVELES.....</i>	127
57.- <i>NÚMERO DE LLAMADAS EN EL CP CON RESPECTO AL TIEMPO DE SIMULACIÓN CON CONTROL A DOS NIVELES.....</i>	128
58.- <i>UTILIZACIÓN DEL CPU.....</i>	129
59.- <i>UTILIZACIÓN DE LDB.....</i>	129
60.- <i>UTILIZACIÓN DE RDB CON EL MECANISMO DE CONTROL DE CONGESTIÓN A DOS NIVELES.</i>	130
61.- <i>RETARDO DE LLAMADAS INTELIGENTES CON CONTROL DE CONGESTIÓN A DOS NIVELES. .</i>	131
62.- <i>DISTRIBUCIÓN DE RETARDOS DE LLAMADAS INTELIGENTES CON CONTROL DE CONGESTIÓN A DOS NIVELES.....</i>	135
63.- <i>GOODPUT EN EL CP CON MECANISMO DE CONTROL DE CONGESTIÓN A DOS NIVELES.....</i>	136
64.- <i>COMPARACIÓN DEL ESTADO DEL BUFFER EN EL CP UTILIZANDO LOS MECANISMOS DE CONTROL BAJO ESTUDIO Y LA CONFIGURACIÓN DE RED INTELIGENTE SIN CONTROL.....</i>	140
65.- <i>COMPARACIÓN DEL GOODPUT EN EL CP SIN CONTROL CONTRA LOS MECANISMOS DE CONTROL ESTUDIADOS.....</i>	141
66.- <i>JERARQUÍA DE MODELOS EN OPNET.....</i>	153

LISTA DE TABLAS

Tabla	<u>Página</u>
I.- <i>RELACIÓN DEL MODELO CONCEPTUAL CON LA IMPLEMENTACIÓN DE LA RED INTELIGENTE.</i>	14
II.- <i>RELACIÓN DE ENTIDADES FÍSICAS CON ENTIDADES FUNCIONALES</i>	19
III.- <i>PARÁMETROS GENERALES UTILIZADOS PARA MODELAR A LA RED INTELIGENTE.</i>	69
IV.- <i>PARÁMETROS PARA EL MECANISMO DE CONTROL CALL GAPPING.</i>	76
V.- <i>RESULTADOS OBTENIDOS CON LA UTILIZACIÓN DE DIFERENTES POLÍTICAS DE INCREMENTO.</i>	80
VI.- <i>RESULTADOS OBTENIDOS CON LA UTILIZACIÓN DE DIFERENTES POLÍTICAS DE DECREMENTO.</i>	81
VII.- <i>PARÁMETROS PARA EL MECANISMO DE CONTROL DE LA VENTANA EN RI.</i>	82
VIII.- <i>PARÁMETROS PARA EL MECANISMO DE CONTROL DE CONGESTIÓN A DOS NIVELES EN RI.</i>	85
IX.- <i>ESCENARIO DE LLAMADAS INTELIGENTES.</i>	90
X.- <i>TABLA COMPARATIVA DE LOS MECANISMOS DE CONTROL Y SIN CONTROL.</i>	144

ESTUDIO DE MECANISMOS DE CONTROL DE FLUJO PARA LA ENTREGA EFICIENTE Y ROBUSTA DE SERVICIOS EN LA RED INTELIGENTE (RI)

I. INTRODUCCIÓN

I.1 ANTECEDENTES DE LA RED INTELIGENTE

Durante los pasados 25 años, las redes telefónicas se han expandido enormemente en términos del número de servicios a subscriptores y del volumen de tráfico transportado. La expansión de los servicios ha sido manejado por la necesidad del incremento sofisticado en los negocios y los subscriptores residenciales. Esta expansión ha permitido el despliegue de la tecnología de Control por Programa Almacenado (SPC, Store Program Control) en las redes (Robrock, 1991), (Audestad, 1993).

La inteligencia proporcionada por la tecnología de Control por Programa Almacenado inicialmente estuvo enfocada a los sistemas de conmutación de la red. El primer sistema utilizando este principio fue desarrollado en el año de 1965 por la AT&T. Este sistema permitía ofrecer servicios residenciales tales como, llamada en espera (call waiting) y servicios a negocios, tal como el Centrex que fue introducido rápidamente.

Durante los años 70, la inteligencia del Control por Programa Almacenado fue empezando a introducirse en sistemas cuyas funciones eran las de soporte al manejo y mantenimiento de la red. Estos sistemas colectivamente conocidos como Sistemas de Soporte de Operaciones (OSS, Operation Support System), fueron necesarios debido al crecimiento y complejidad de las redes y servicios ofrecidos a los subscriptores.

Al comienzo de 1981, la inteligencia de las redes alcanza una nueva etapa, cuando la AT&T introduce el uso de bases de datos centralizadas para introducir los servicios: llamada por tarjeta (calling card) y el número 800 (llamada gratuita). Una o varias bases de datos se ubican sobre los sistemas de Punto de Control de Red (NCP, Network Control Point) y son accesadas por los sistemas de conmutación a través de la red de Señalización Interoficina de Canal Común (CCIS, Common Channel Interoffice Signaling network). Esta centralización permite la introducción de algunos servicios que de otro modo serían imprácticos debido a la complejidad para manejar grandes cantidades de datos volátiles en cada conmutador SPC.

Después de la diversificación del sistema Bell a mediados de los 80's, emergieron varias compañías regionales que desarrollaron bases de datos centralizadas en los llamados Puntos de Control de Servicio (SCP, Service Control Point), para tolerar Servicios de Facturación Alternativa (ABS, Alternate Billing Service) y el servicio 800. ABS provee validación de tarjetas telefónicas y otras funciones de información en línea para el cobro y tarificación a terceras personas. El acceso al SCP desde conmutadores es proporcionado a través de la red del sistema de señalización número 7 (SS7, Signaling System number 7). La funcionalidad en los conmutadores (SSPs, Service Switching Points), redes SS7, SCP y OSS que asisten a estos servicios es colectivamente conocida como **Red Inteligente (RI: IN Intelligent Network)**, (Berman y Brewster, 1992).

Reconociendo que había un gran potencial en el número de servicios que podrían ser ofrecidos mediante la expansión de la funcionalidad de la RI; se realizaron esfuerzos de expansión en Bellcore a petición de las compañías regionales para definir una arquitectura que pudiera facilitar la introducción de nuevos servicios. El resultado de estos esfuerzos

llevaron a la realización de un plan para una arquitectura conocida como Red Inteligente 2 (IN/2, RI/2). Esta arquitectura incluyó un gran conjunto de capacidades expandidas en el conmutador y en el SCP, conocidas como Componentes Funcionales (FC, Functional Component), y un nuevo sistema llamado Periféricos Inteligentes (IP, Intelligent Peripheral). Los FCs definieron los elementos atómicos funcionales de la arquitectura y los IP proporcionaron una plataforma para desarrollar capacidades de servicios-asistencia. Esta arquitectura pretendió apoyar una gran cantidad de servicios de voz y datos, y ser programable por el operador telefónico, pudiendo facilitar el crecimiento de esos servicios.

Basado en análisis realizados por compañías regionales y vendedores de conmutadores, se determinó que RI/2 fue una propuesta demasiado ambiciosa, la cual podría conllevar altos riesgos y no podría ser implementada en un tiempo suficientemente corto. La atención fue entonces enfocada al diseño de una arquitectura con un subconjunto de la funcionalidad de RI/2 la cual pudiera introducir la capacidad de servicios independientes a la red en pocos años. La arquitectura resultante fue conocida como RI/1+ (Baver et al, 1988). Similar a RI/2, incluía el uso de FCs y Periféricos Inteligentes, pero ellos estaban limitados a un pequeño conjunto de funciones necesarias para asistir en servicios de voz. Aunque las reacciones de los vendedores de conmutadores a la RI/1+ generalmente fueron de apoyo, se podía prever que una arquitectura basada en SCP/SS7 no podría mantener todos los servicios deseados con un suficiente nivel de desempeño. También era previsible que se podrían lograr mejores interacciones entre la arquitectura RI y la arquitectura de la red de conmutadores. Basados en estos factores, se decidió detener el esfuerzo RI/1+ y formar un foro llamado interacciones entre multivendedores (MVI, Multi-

Vendors Interactions), incluyendo a Bellcore, las compañías regionales y la comunidad de vendedores, con el propósito de recabar las mejores ideas de todas estas fuentes y definir una arquitectura que podría lograr a largo plazo los objetivos de las compañías regionales y a su vez tener el soporte de los vendedores. El foro MVI fue lanzado al comienzo de 1989 con dieciséis compañías regionales. El número de vendedores participantes creció rápidamente a 22. El resultado técnico de los procesos MVI fueron publicados en marzo de 1990.

La siguiente etapa de la evolución de la RI comenzó por MVI y fue llamada red inteligente avanzada (AIN, RIA), (Berman y Brewster, 1992). La arquitectura RIA fue diseñada desde un conjunto de necesidades funcionales asociadas con la provisión de servicios de telecomunicaciones de voz. Hoy en día, la red inteligente es la dorsal que permite que nuevos servicios en Telecomunicaciones sean agregados en una forma rápida, económica y eficiente.

La principal característica de la arquitectura RIA es su flexibilidad con respecto a su implementación física. Los sistemas físicos e interfaces incluidas en la arquitectura de RIA se ilustran en la figura 1

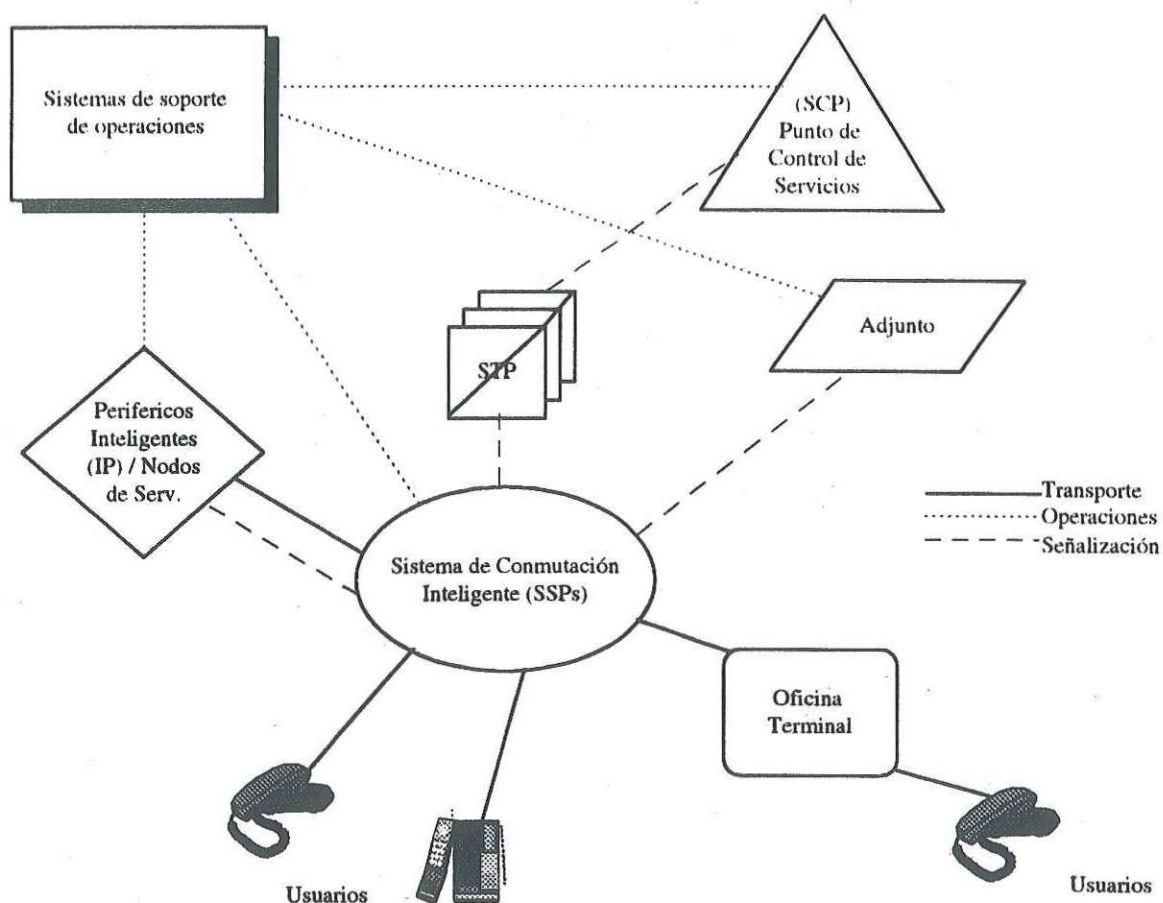


Figura 1 Arquitectura de la RIA

No todos los sistemas identificados como componentes de la arquitectura RIA necesitan ser desarrollados para proveer las capacidades de los servicios de RI. La decisión respecto a qué elementos de la red deberían desarrollarse dependerá de una variedad de factores, incluyendo las características existentes de la red y los planes de desarrollo, los tipos de servicios a ser ofrecidos, y la similitud entre los servicios a ser desarrollados.

Los servicios de RI son proporcionados a través de interacciones entre sistemas de conmutación y sistemas de soporte de la lógica de servicio de RI. Los sistemas de conmutación detectan condiciones en las cuales pueden estar involucrados un servicio de RI

(es decir, estos encuentran un “disparador”), formulan una petición de servicio inteligente, y responden a instrucciones de procesamiento de llamada desde los elementos de red en los cuales reside la lógica de los servicio inteligentes.

Uno o mas vehículos para proporcionar la lógica de servicio inteligente están disponibles en la red. Dependiendo de las consideraciones de implementación y el servicio a ser ofrecido, la lógica del servicio inteligente puede ser proporcionado en el SCP, el sistema adjunto, los nodos del servicio, o en una plataforma programable dispuesta en el sistema de conmutación.

I.2 PLANTEAMIENTO DEL PROBLEMA

El SCP es el elemento más importante en la arquitectura de RI. Puesto que el SCP funciona como un servidor lógico conteniendo la mayoría de la información y la lógica de servicio. En un futuro muy cercano, la introducción de nuevos y mas avanzados servicios de RI implicarán no tan solo un incremento en el tráfico de señalización, sino también un incremento en la carga de tráfico impuesta al SCP . Por lo tanto, ésto provocará problemas de cuellos de botellas en los nodos centralizados, lo cual puede llevar a un incremento inaceptable en el tiempo de respuesta para una llamada inteligente y a la vez, reducir la capacidad de manejo de llamadas en la red, es decir, la reducción del caudal eficaz en el sistema.

Con el fin de aliviar el problema de cuello de botella en el sistema, e incrementar el caudal eficaz de llamadas inteligentes y a la vez disminuir el tiempo de respuesta. Hacemos uso de una configuración para el SCP en la cual múltiples procesadores de llamadas (CPs,

Call Processors) compuestos de una base de datos local (LDB, Local DataBase) cada uno de ellos, participan en el procesamiento de llamadas inteligentes. Estos procesadores de llamadas están enlazados uno a otro por medio de una red dual Token Ring de alta velocidad, compartiendo una base de datos remota (RDB, Remote DataBase) como se muestra en la figura 2. El uso de múltiples procesadores de llamadas tendrá el efecto de incrementar el caudal eficaz de llamadas inteligentes y a la vez garantizar el proceso de una llamada si un procesador de llamada falla.

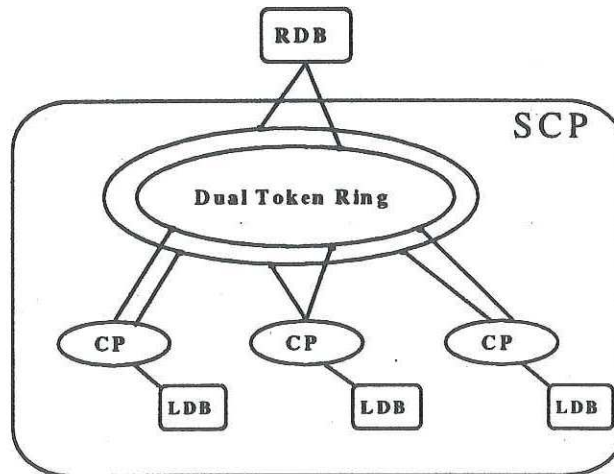


Figura 2 Arquitectura multiprocesadores para el SCP propuesta.

De igual manera, hemos propuesto el estudio de mecanismos de control de flujo con el fin de controlar el flujo de tráfico impuesto a los procesadores de llamadas que componen a un SCP. Diferentes métodos han sido evaluados con este fin, pero ahora nosotros utilizamos una nueva arquitectura de multiprocesadores en el SCP, haciendo uso del control de flujo Call Gapping y Window. Además hemos propuesto un nuevo mecanismo de control

de congestión a dos niveles que nos permite controlar el uso de la base de datos remota. El uso de la base de datos remota es propuesta por dos razones. Primero para compartir la carga, esto se refiere al caso en que cuando el número de llamadas en espera de ser servidas por los procesadores de llamadas exceden la capacidad de servicio de la base de datos local. Segundo por confiabilidad, refiriéndonos en este caso a que una llamada inteligente puede continuar siendo atendida aunque una de las base de datos local falle.

I.3 OBJETIVOS GENERALES

El principal objetivo de nuestro estudio es evaluar el uso de tres mecanismos de control implementados en una arquitectura de multiprocesadores que conforma el núcleo de la Red Inteligente con el fin de proteger al SCP de sobre flujo de tráfico, el cual decrementa el caudal de llamadas inteligentes procesadas en el SCP e incrementa el tiempo de respuesta de las mismas. Para cumplir con este objetivo principal es necesario cubrir los siguientes sub-objetivos:

- Estudiar la configuración básica de una Red Inteligente, compuesta por entidades físicas como el SSP , SCP y la red de Señalización (STPs).
- Estudiar en detalle una nueva arquitectura interna de multiprocesadores para el SCP con el fin de incrementar el caudal de llamadas inteligentes.
- Estudiar el sistema usando un escenario de llamadas inteligentes en el cual se capture una variedad de servicios inteligentes a varios niveles de complejidad. Considerando el uso futuro de la red inteligente.
- Evaluación del sistema vía técnicas de simulación implementando modelos de colas en

OPNET (ver apéndice A), haciendo uso de diferentes mecanismo de control con el fin de cumplir con los requerimientos de calidad de servicio en la red inteligente como son:

- Tiempo de Respuesta (El 95% de las llamadas admitidas deben de ser cumplidas en menos de 2 segundos).
- Probabilidad de Bloqueo (Un máximo del 5% del número de llamadas generadas).

I.4 ORGANIZACIÓN DE LA TESIS

En este capítulo hemos introducido las características básicas del funcionamiento de la red inteligente, como también hemos explicado nuestro caso de estudio y los objetivos generales a cubrir con nuestro trabajo. En el capítulo II se estudia la funcionalidad de la Red Inteligente como un modelo conceptual y como una arquitectura física, y se presentan el conjunto de capacidades desarrolladas por la ITU-T, con el propósito de estudiar los tipos de servicios que pueden ser implementados en la Red Inteligente.

El capítulo III presenta una breve descripción de los conceptos de control de flujo y congestión, y se describe el funcionamiento de los mecanismos de control en la red inteligente, propuestos para ser estudiados e implementados en un modelo vía técnicas de simulación. El capítulo IV nos muestra la configuración de red inteligente diseñada para nuestro estudio usando el simulador OPNET. También se explica como se implementaron los mecanismos de control y que efectos tienen en la RI. En el capítulo V se analizan e interpretan los resultados de simular nuestra configuración de red inteligente sin control y bajo tres diferentes mecanismos de control.

El último capítulo (VI), contiene las conclusiones, contribuciones y comentarios de nuestro trabajo de tesis. El Apéndice A contiene una breve descripción del método de simulación utilizado, en donde se describen las características principales de OPNET.

II. RED INTELIGENTE

II.1 INTRODUCCIÓN

La RI es un concepto de arquitectura que nos permite de manera flexible, rápida y económica el diseño y desarrollo de servicios novedosos en redes de telecomunicaciones existentes y futuras. Con el fin de disponer de estos servicios con una mínima modificación en la red, la introducción de nuevos servicios deberá ser de una manera transparente al usuario y proveedor de la red. En la red inteligente, la introducción de nuevos servicios dejan de estar dependiendo del hardware y software disponible en los conmutadores de control por programa almacenado. Ahora bien, mucha de la inteligencia está centralizada en bases de datos las cuales son accedadas a través de la red de señalización de conmutación de paquetes. Con este concepto el control de la llamada se lleva a cabo entre el Punto de Conmutación de Servicios conocido como SSP y las bases de datos centralizadas en el Punto de Control de Servicios llamado SCP.

En este capítulo, se estudiará el concepto de Red Inteligente desde dos puntos de vista, desde un Modelo Conceptual de Red Inteligente basado en las series de recomendaciones de la ITU-T Q.120x y el modelo físico de la arquitectura de Red Inteligente, ésto con el fin de distinguir las funciones e implementaciones de la RI. Del mismo modo estudiaremos el uso de la red de señalización conocida como SS7, constituida como la dorsal de comunicación en la Red Inteligente. Por último para finalizar este capítulo presentaremos la evolución del conjunto de capacidades de la red inteligente. En donde un conjunto de capacidades se refiere al conjunto de servicios y sus características

que pueden ser contruidos usando bloques de servicios independientes para cada etapa evolutiva de la red inteligente.

II.2 MODELO CONCEPTUAL DE RED INTELIGENTE (INCM)

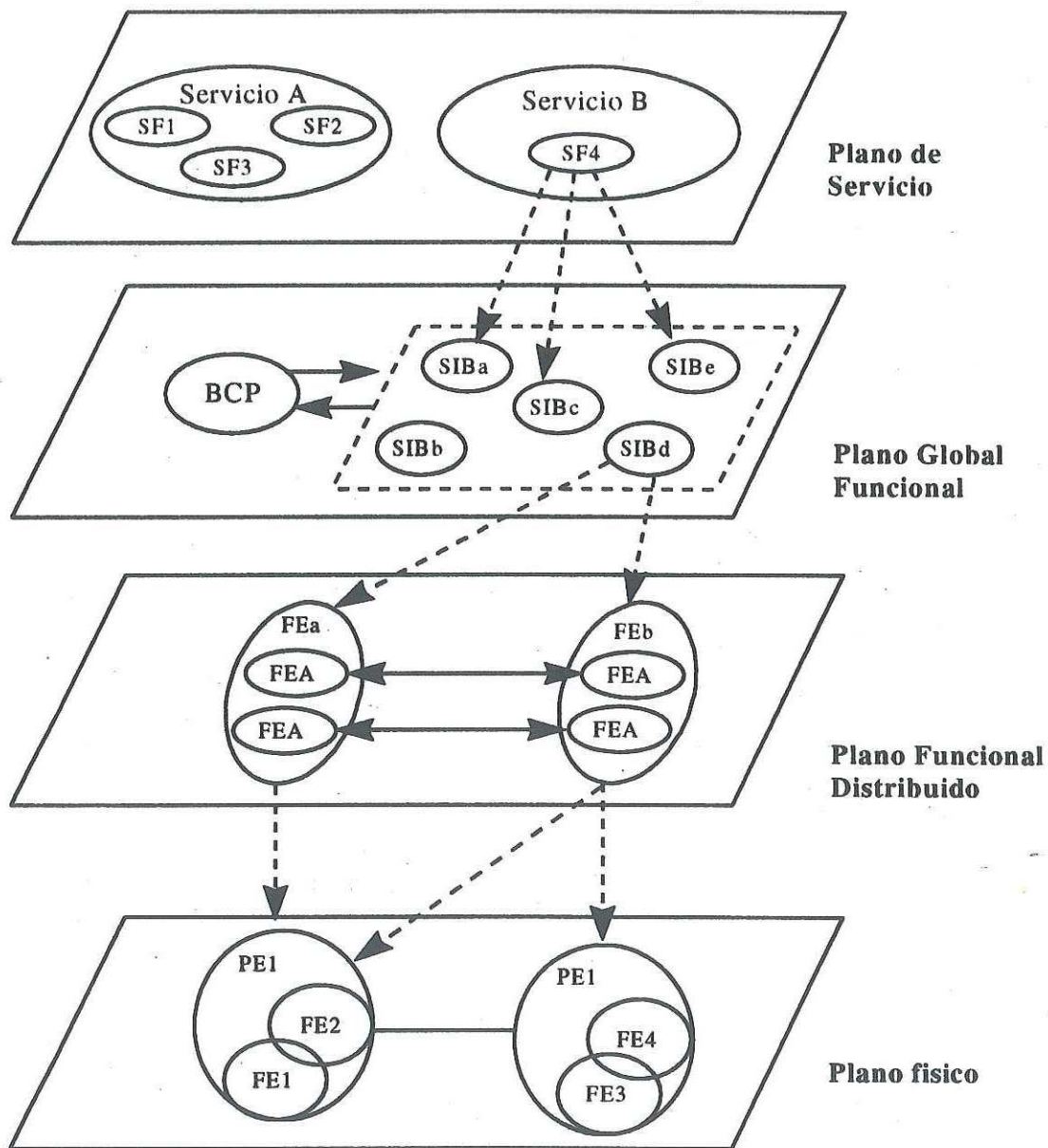
El modelo conceptual de red inteligente describe la arquitectura de la red en términos de las relaciones entre los componentes funcionales de la RI y la descomposición lógica de los servicios inteligentes en forma de una estructura jerárquica. El principal objetivo de un modelo conceptual de RI es el de distinguir las funciones de la red inteligente de la implementación misma de la red. El modelo de red inteligente fue creado por ETSI (European Telecommunication Standars Institute) y la ITU-T (International Telecommunication Unión - Telecommunication) y es descrita en las series de recomendaciones Q.120x por la ITU-T. Los objetivos principales del modelo conceptual de RI son:

- Implementación independiente de servicios.
- Implementación independiente de la red.
- Independencia de vendedores.
- Independencia de tecnología.

El modelo conceptual de red inteligente se muestra en la figura 3, el cual consiste de una estructura jerárquica compuesta por cuatro planos, que son:

- Un Plano de Servicio
- Un Plano Global Funcional (GFP, Global Functional Plane)

- Un Plano Funcional Distribuido (DFP, Distributed Functional Plane)
- Un Plano Físico



SF : Características de Servicio
BCP: Proceso Básico de Llamada
PE: Entidades Físicas

SIB: Bloques de Servicios Independientes
FE: Entidades Funcionales

Figura 3 Modelo Conceptual de Red Inteligente (ITU-T Rec. Q.1201, 1992)

La relación que existe entre el modelo conceptual y la actual implementación de la Red Inteligente puede ser vista en la tabla I.

Tabla I Relación del Modelo Conceptual con la Implementación de la Red Inteligente.

Modelo Conceptual	Implementación	Descripción
<i>Plano de Servicio</i>	Servicios	El plano que los usuarios ven.
<i>Plano Global Funcional</i>	SIBs	Bloques de Servicios Independientes.
<i>Plano Funcional Distribuido</i>	Entidades Funcionales	Funciones específicas en la red.
<i>Plano Físico</i>	Productos Físicos	Entidades Físicas en la red, tales como SSP, SCP, etc.

II.2.1 EL PLANO DE SERVICIO

El plano de servicios está descrito en las recomendaciones de la ITU-T Q.1202 (Q.1202, 1993). Este plano provee un punto de vista exclusivamente orientado a servicios. Es decir, muestra los servicios a los usuarios sin indicarles como han sido implementados en la red. El plano de servicio está formado por pequeñas funciones llamadas “características de servicios” (SFs, Service Features). En donde un servicio puede estar compuesto por varias SFs. Las SFs son mayores que los bloques independientes de servicios (SIBs, Service Independent Block) pero menores que un servicio (Figura 3, Plano de Servicio).

Por otra parte, el plano de servicios contiene la personalización y administración de servicios. Un importante problema a considerar en este plano son las interacciones entre las características de los servicios. Es decir, cuando un efecto no esperado ocurre en el

momento que dos o mas servicios o SFs son utilizados al mismo tiempo. Los SFs del plano de servicio son relacionados en GFP por medio de la combinación de SIBs en el GFP usando la lógica de servicio global (GSL, Global Service Logic).

II.2.2 EL PLANO GLOBAL FUNCIONAL (GFP)

El plano global funcional está descrito en la recomendación de la ITU-T Q.1203 (Q.1203, 1994). El GFP es el plano en donde se encuentran los SIBs. También en este plano se encuentran los operadores de red, los cuales crean nuevos servicios básicos. Los SIBs son pequeños bloques independientes de servicios utilizados para formar un servicio en particular. Estos pueden ser utilizados varias veces con el fin de crear SFs.

El GFP modela la funcionalidad de la red desde un punto de vista global. En este plano, servicios y características de servicios son definidos en términos de las funciones de red requeridas para soportarlos. Estas funciones no son servicios, ni características específicas de servicio sino son considerados como SIBs.

En este plano se encuentra el proceso básico de llamada (BCP, Basic Call Procedure) el cual es un SIB especial que maneja todas las actividades necesarias para una llamada normal. Un SF puede estar compuesto por uno o mas SIBs; los SIBs están conjuntamente encadenados por un GSL. Por lo tanto, GSL es el único elemento en GFP que es dependiente de servicio. Un SIBs en el GFP puede encontrarse en al menos una entidad funcional (FE, Functional Entity) en el DFP, pero también puede ser encontrado en más de un FE.

II.2.3 EL PLANO FUNCIONAL DISTRIBUIDO (DFP)

El DFP, está descrito en la recomendación de la ITU-T Q.1204 (Q.1204, 1993). DFP provee la distribución de funciones en la red inteligente. Éste contiene FE's, sus respectivas acciones (FEAs, Functional Entity Actions) y el flujo de información entre las últimas. Cada FE describe una funcionalidad específica en la RI la cual es físicamente implementada en la red. Por ejemplo: el control de llamadas, conmutación de servicio y el control del servicio. Por ésto, el DFP es el principal interés de los proveedores de red y diseñadores. La arquitectura del plano funcional distribuido se muestra en la figura 4.

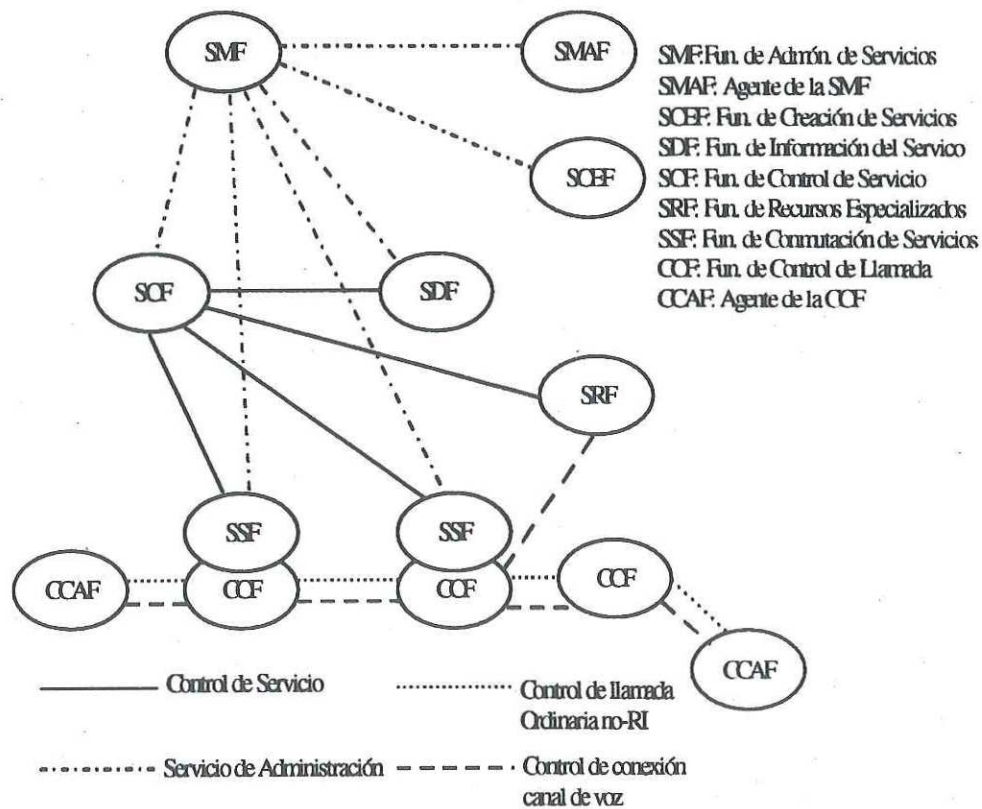


Figura 4 Plano Funcional Distribuido.

De la figura tenemos que:

Función de Control de Llamada (CCF, Call Control Function)

Esta función maneja todas las llamadas normales y la conmutación de llamadas y servicios, ambos basados en llamadas inteligentes o no inteligentes.

Agente de la Función de Control de llamada (CCAF, Call Control Agent Function)

Esta función maneja el acceso de los usuarios a los servicios. Esta función reside en los conmutadores locales.

Función de conmutación de servicios (SSF, Service Switching Function)

Esta función determina cuando una lógica de servicio inteligente debe de ser invocada. El SSF es responsable de conmutar una llamada o servicio a un lugar en particular basado en las instrucciones obtenidas de la función de control de servicios.

Función de control de servicio (SCF, Service Control Function)

Esta función constituye el núcleo de la red inteligente. Esta función controla todos los procesos del manejo de llamadas o servicios enviando instrucciones al SSF/CCF, a la función de información de servicios (SDF, Service Data Function), y a la función de recursos especializados (SRF, Service Resource Function). La lógica de servicios y la información asociada se encuentra únicamente en el SCF.

Función de información de servicios (SDF)

Esta función asiste al SCF con información acerca del cliente y de la red.

Función de recursos especializados (SRF)

Esta función juega el papel de interfaz entre la entrada de datos de usuario y el SCF.

El SRF colecciona datos de usuarios.

Función de creación de servicios (SCEF, Service Creation Environment Function)

La tarea de esta función es el definir, desarrollar y proveer servicios inteligentes.

Función de administración de servicios (SMF, Service Management Function)

Esta función virtualmente se conecta con todas las otras funciones en la red. Se encarga de manejar la distribución, control y mantenimiento de un servicio inteligente.

Agente de la función de administración de servicios (SMAF, Service Management Agent Function)

Esta función opera como una interfaz entre operadores de mantenimiento y el SMF.

Las FEs están localizadas en las entidades físicas (PEs, Physical Entity) en el plano físico.

II.2.4 EL PLANO FÍSICO

El plano físico está descrito en la recomendación de la ITU-T Q.1205 (Q.1205, 1996), el cual determina el aspecto físico de la arquitectura de Red Inteligente. Este plano identifica las diferentes PEs o productos reales y protocolos que pueden existir en una estructura real de Red Inteligente. También, en este plano se indica que entidades funcionales componen a una entidad física.

Las entidades físicas pueden ser definidas como los elementos o dispositivos computarizados en una red estructurada de servicios inteligentes. Las entidades físicas que componen el plano físico en el modelo conceptual de Red Inteligente se muestran en la figura 5, como son:

- Punto de Acceso a la Red (NAP)

- Punto de Conmutación de Servicios (SSP)
- Punto de Conmutación y Control de Servicios (SSCP, Service Switching Control Point)
- Punto de Control de Servicios (SCP)
- Punto de Información de Servicios (SDP, Service Data Point)
- Sistema de Administración de Servicios (SMS, Service Management System)
- Sistema de Creación de Servicios (SCE, Service Creation Environment)
- Periféricos Inteligentes (IP)
- Adjuntos (AD, Adjunct)
- Nodos de Servicios (SN, Service Node)

Cada entidad físicas contiene al menos un entidad funcional. La tabla II, muestra la relación entre varias entidades físicas con sus respectivas entidades funcionales.

Tabla II Relación de Entidades Físicas con Entidades Funcionales

<i>Entidades Físicas</i>	<i>Entidades Funcionales</i>							
	CCAF	CCF	SSF	SCF	SDF	SRF	SMF	SCEF
NAP	C	O	-	-	-	-	-	-
SSP	O	C	C	O	O	O	-	-
SCP	-	-	-	C	O	-	-	-
SDP	-	-	-	-	C	-	-	-
SSCP	O	C	C	C	C	O	-	-
SMS	-	-	-	-	-	-	C	O
SCE	-	-	-	-	-	-	-	C
AD	-	-	-	C	C	-	-	-
IP	-	O	O	-	-	C	-	-
SN	-	C	C	C	C	-	-	-

C= Función Necesaria, O= Función opcional, - = Función no permitida.

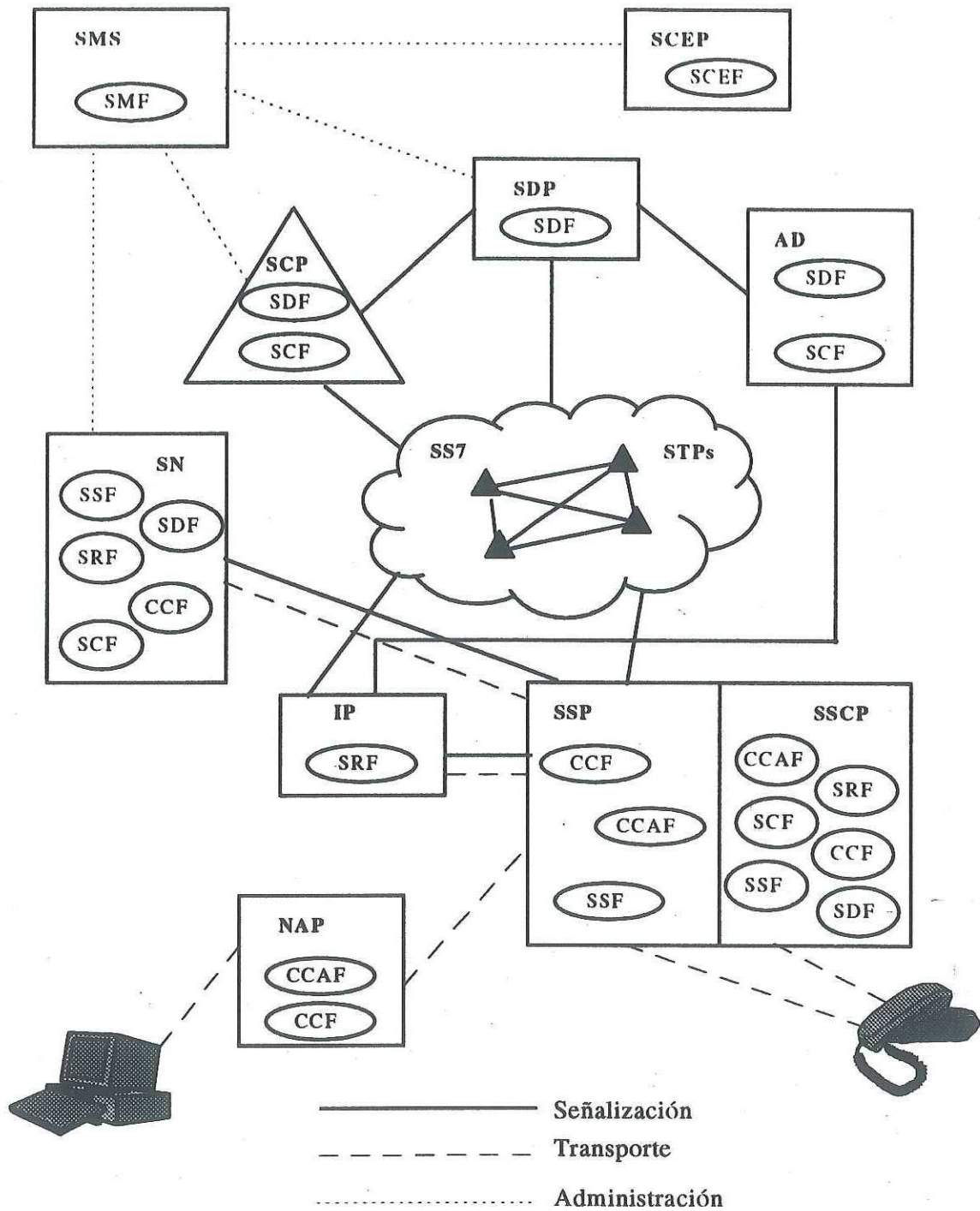


Figura 5 *Arquitectura Física de la Red Inteligente basada en el Sistema de Señalización número 7 (SS7)*

II.3 ARQUITECTURA FÍSICAS DE LA RED INTELIGENTE

La arquitectura de red inteligente está formada por dos aspectos principales :

- 1) basada en la red de telecomunicaciones de señalización por canal común, con el objetivo de ofrecer un intercambio confiable de mensajes entre las diferentes entidades que conforman la red.
- 2) la red inteligente nos permite la rápida creación de servicios e implementación de los mismos.

Por lo tanto, la estructura de red de telecomunicaciones inteligente está constituida por:

- Una red de señalización, la cual consiste de :
 - Enlaces de Señalización.
 - Puntos de Transferencia de Señalización.
 - Puntos de Señalización como:

• NAP,	• SSCP,	• AD,
• SSP,	• SDP,	• SN.
• SCP,	• IP,	
- No-señalización, sistemas relacionados a la creación de servicios y administración de la red, los cuales pueden consistir de:
 - Funciones de administración y creación de servicios, como:
 - SMS,
 - SCE.

La figura 5 nos muestra la arquitectura física de la red inteligente y la relación de cada una de las entidades físicas con respecto a las entidades funcionales que la componen. También se muestra el uso de la red de señalización como dorsal de comunicación entre las diferentes entidades físicas. Los puntos de servicios mas importantes que integran la arquitectura de Red Inteligente son descritos a continuación.

II.3.1 PUNTO DE CONMUTACIÓN DE SERVICIO (SSP)

El SSP es un conmutador en la Red Telefónica Pública Conmutada (PSTN, Public Switched Telephony Network), el cual ha sido modificado con el fin de reconocer un servicio inteligente. En la red inteligente actual, una petición de servicio durante una llamada normal es detectada por medio de un punto de conmutación, el cual identifica el servicio inteligente requerido. El SSP establece una conexión con el SCP, en donde la lógica de servicio se lleva a cabo. El SSP no transfiere el circuito de voz hacia el SCP; pero usa un protocolo llamado TCAP (Transaction Capabilities Application Part) o INAP (IN Application Protocol) para dialogar con el SCP. El SSP está compuesto por tablas de enrutamiento que permiten determinar cual SCP y cuales servicios están disponibles en la red SS7.

Además de proveer el acceso a los usuarios a la red inteligente, el SSP desempeña cualquier función necesaria por un conmutador, siempre y cuando el SSP se use como un conmutador local. El SSP también tiene la capacidad de comunicarse con otras entidades físicas conteniendo una función de control de servicio tal como un SCP, y pudiendo responder a instrucciones de otras entidades físicas, con ésto, el SSP se convierte en un

punto de conmutación y control de servicio mejor conocido como SSP. Funcionalmente, un SSP contiene CCFs, la SSF, y si el SSP es usado como conmutador local, éste contiene CCAF (Q.1205, 1996). El SSP está compuesto de los siguientes componentes, tal y como lo muestra la figura 6:

- Partes de aplicación las cuales definen los servicios que son soportados.
- Una TCAP, la cual provee una parte de la capa del protocolo de aplicación.
- Una capa de red, la cual define la red de comunicación por la cual los protocolos son diseñados. El SCCP está actualmente definido como la capa de red, basado en el SS7 MTP (Transferencia de mensaje de señalización).

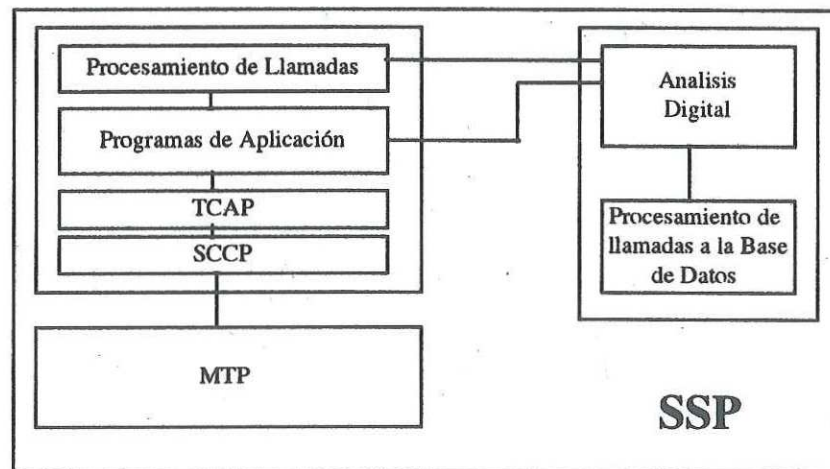


Figura 6 Componentes del SSP.

II.3.2 PUNTO DE CONTROL DE SERVICIOS (SCP)

La principal tarea del SCP es proporcionar un acceso altamente confiable a, y procesamiento de, aplicaciones de bases de datos. Este nodo centralizado es el núcleo de la red inteligente el cual controla toda la inteligencia en la red. En lo que respecta a

funcionalidad, el SCP contiene la SCF, usualmente una SDF, y la lógica de servicio conocida como programas de lógica de servicios (SLP). El SCP debe asignar sus recursos para realizar las siguientes funciones:

Procesamiento de mensajes:

Esta comprende aquellas funciones requeridas para entregar un mensaje a su destino correcto. Estas funciones incluyen:

- Ejecución de protocolos de red como CCS7, X.25.
- Manejo de los errores de protocolo.
- Discriminación de mensajes.
- Distribución de mensajes y enrutamiento.
- Prueba y administración de la red.
- Procesamiento de aplicaciones.
- Administración del nodo.

Operaciones de nodo, administración y mantenimiento:

Estas funciones son requeridas para controlar y operar al SCP :

- Colección y reporte de medidas de desempeño.
- Seguridad.
- Detección de errores y fallas, notificación, aislamiento y recuperación.
- Control de flujo y detección de sobrecarga.
- Monitoreo del estado y bienestar de la red.
- Control de la configuración.

El SCP proporciona soporte a un gran volumen de transacciones de bases de datos en un ambiente de tiempo real, multitarea y de alta disponibilidad. Para una disponibilidad de red extremadamente alta, los SCPs y STPs pueden ser dispuestos geográficamente en topologías apareadas. Para una máxima flexibilidad, es medular el incluir en la arquitectura del SCP los siguientes cuatro componentes principales, tal y como se muestra en la figura 7:

- **Conectividad**

Proporciona comunicación confiable entre los otros tres componentes (Terminal frontal, Nodo Manejador y Terminal posterior) usando una arquitectura doble Token Ring para mayor velocidad, disponibilidad, capacidad y flexibilidad. Cada uno de esos tres componentes consisten de subsistemas de proceso atados al anillo, proporcionando capacidad y disponibilidad apropiada. Para cada componente se configuran réplicas de procesadores con software y hardware similares para uniformidad e intercambiabilidad. El método de réplica de procesadores es operado normalmente en una manera de balance de carga para desempeño óptimo.

- **Terminal Frontal**

Estos componentes proporcionan las interfaces externas de comunicaciones a las redes de soporte y servicio. Cada interfaz externa se trata independientemente, así que hay actualmente una serie de terminales frontales para un SCP. Cada ambiente puede ser diseñado para proporcionar solo aquellas interfaces relevantes a las redes accedidas desde ese escenario, y conservando la flexibilidad para incrementar el número de interfaces soportadas en el futuro. Los requerimientos de las interfaces externas son:

- a).- La interfaz CCSN (MTP niveles 1-3 y SCCP)
- b).- La interfaz X.25
- c).- La interfaz de la red de los sistemas operativos.

- **Terminal Posterior**

Los componentes de la terminal posterior son una base para los servicios de aplicación de las bases de datos y consisten de procesadores manejadores de las aplicaciones de las bases de datos. Los procesadores son consultados como procesadores anfitriones de aplicaciones (AHP).

- **Nodo Manejador**

Los componentes del nodo manejador proveen un punto de coordinación centralizada para el nodo SCP. El manejo completo del nodo incluye todas las funciones de coordinación y control requeridas para asegurar la continuidad de las operaciones, administración y mantenimiento de los escenarios que se lleven a cabo en la red.

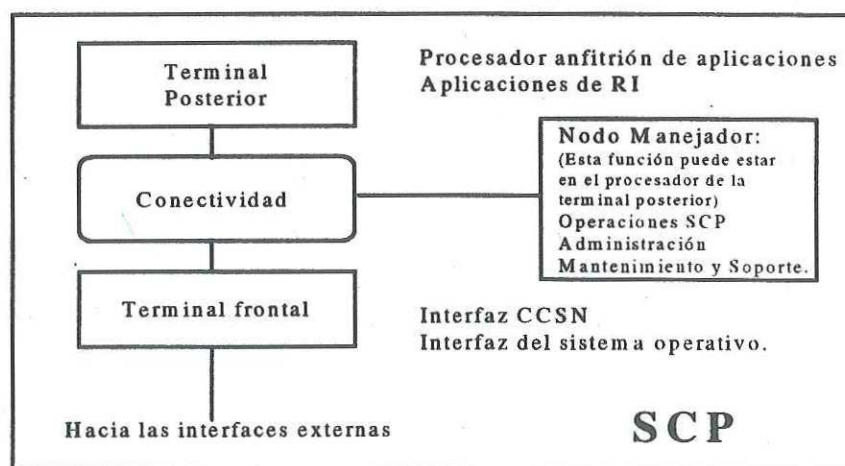


Figura 7 Componentes del SCP.

II.3.3 PUNTOS DE TRANSFERENCIA DE SEÑALIZACIÓN (STP)

Los STP son enrutadores de paquetes que componen a la red SS7. Estos son usados para permitir el dialogo entre diferentes entidades físicas tales como el SSP con el SCP. Por lo tanto, el STP debe de implementar o suministrar las siguientes características:

- Unidad de Mensaje de datos (UDT, Unit Data Transfer), para transferir información para el usuario.
- Unidad de servicio de mensajes de datos (UDTS, UDT Service), para reportar una entrega no exitosa de un UDT a un punto de señalización original es decir a una entidad física.
- Traducción de títulos globales (GTT, Global Translate Title), para trasladar algunos tipos de direcciones globales, tales como un servicio de disparo inteligente, por un destino apropiado.

Además, el STP sirve como un concentrador de tráfico para SCPs los cuales pueden ser agregados donde la red de señalización SS7 no está bien constituida para manejar el caudal eficaz adicional de la red inteligente.

II.3.4 PERIFÉRICOS INTELIGENTES (IP)

Los IPs proveen recursos optimizados para la personalización de servicios, y soportan una comunicación flexible de información entre usuarios y la red. Algunos recursos usados en los IPs son:

- anuncios de voz,

- reconocimiento de voz,
- generador de tonos,
- puentes para audioconferencias,
- conversión de protocolos y,
- colección de dígitos usando DTMF (Dual Tone Multiple Frequency).

Los IP contienen una matriz de conmutación para conectar los usuarios a estos recursos.

En su arquitectura funcional el IP incluye al SRF, y opcionalmente funciones de SSF/CCF. Estas funciones opcionales son usadas para proveer acceso externo para conectarse a los recursos dentro del IP. El IP conecta a uno o mas SSPs, y/o a la red de señalización. Un SCP puede requerir el uso de los recursos de un IP, por medio de un SSP que tiene conectado un IP para llevar a cabo un servicio. Del mismo modo el SCP puede requerir el conectar a un usuario con un IP de un SSP, aunque el IP este conectado a otro SSP.

II.3.5 SISTEMA DE ADMINISTRACIÓN DE SERVICIOS (SMS)

El SMS es un sistema de soporte de operaciones utilizado para la administración de servicios inteligentes. El SMS puede ser llamado Punto de administración de servicio (SMP), algunas de sus funciones son:

- Control del manejo de servicios,
- control de implementación del servicio,
- administración de la base de datos,

- prueba de la red,
- administración y manejo del tráfico en la red, y
- tiene la función de coleccionar información acerca del desempeño de la red.

Funcionalmente, el SMS contiene al SMF y, opcionalmente, SMAF y SCCF. El SMS puede ser alcanzado por todas las entidades físicas de la red inteligente. El SMS hace posible el manejo de la información del servicio en un sistema de red inteligente distribuido, y asegura la consistencia de los datos tanto en las bases de datos en el SCP o SDF. Los protocolos comúnmente utilizados para el manejo y administración de la información de los servicios inteligentes son SNMP o CMIP a través de enlaces X.25.

II.3.6 PUNTO DE CREACIÓN DE SERVICIOS (SCEP)

El SCEP (Service Creation Environment Point) es usado para definir, desarrollar, y probar servicios inteligentes, y después implementarlos en el SMS. Funcionalmente, este contiene al SCEF. El SCEP trabaja directamente con el SMS (Q.1205, 1996).

La programación de un servicio inteligente es básicamente basado en la combinación de SIBs. Por lo tanto, el SCEP usa interfaces gráficas llamadas GUI (Graphical User Interface), tales como XWindows, o Microsoft Windows. La descripción de la lógica de servicios puede ser hecho usando métodos formales tales como: Especificación y descripción de lenguaje (SDL, Specified Description Language) o Autómatas de estados finitos extendidas (EFSA, Extended Finite State Automata). Después de todo el diseño y modelado del servicio (funciones de administración y servicios de bases de datos) pueden implementarse usando herramientas CASE (Computer Aided Software Engineering) y/o

DDL (Data Definition Language).

II.4 SISTEMA DE SEÑALIZACIÓN NÚMERO 7 (SS7)

El sistema de señalización número 7 puede ser definido como el sistema que permite que los SPCs, redes de bases de datos, y otros nodos inteligentes de la red intercambien mensajes relacionados al establecimiento de llamadas, supervisión, control de información de conexión, procesos entre pregunta/respuesta, información de usuario e información de administración de red, etc. (Modarresi y Skoog, 1990). El sistema de señalización número 7 o SS7 descrito por la ITU-T (comúnmente llamado sistema de señalización por canal común número 7 o CCS7) transporta información de señalización sobre una red de señalización digital dedicada, separada de los canales de voz. Además de que SS7 permite la separación de los conmutadores inteligentes del enrutamiento de voz y datos en las redes de telecomunicaciones. SS7 también ofrece los siguientes beneficios:

- Mejora en la velocidad y flexibilidad del establecimiento de llamada.
- El rápido intercambio de información entre procesos de una llamada que requiere un enrutamiento y manejo especial.
- Se puede tener acceso a información almacenada en una red de bases de datos la cual permite que compañías desarrollen servicios avanzados para toda la red.
- Con la señalización bi-direccional, SS7 provee al conmutador de origen o al cliente todos los detalles acerca del progreso e información del procesamiento de la llamada, cuando ésta es establecida por la red inteligente.

Como el modelo de referencia OSI, el protocolo SS7 puede estar dividido en diferentes capas de funciones. En el caso de SS7 las capas son llamadas niveles (Q.700, 1988). La comparación del modelo de referencia OSI creado en 1982 con el modelo SS7 creado en 1980 se muestra en la figura 8.

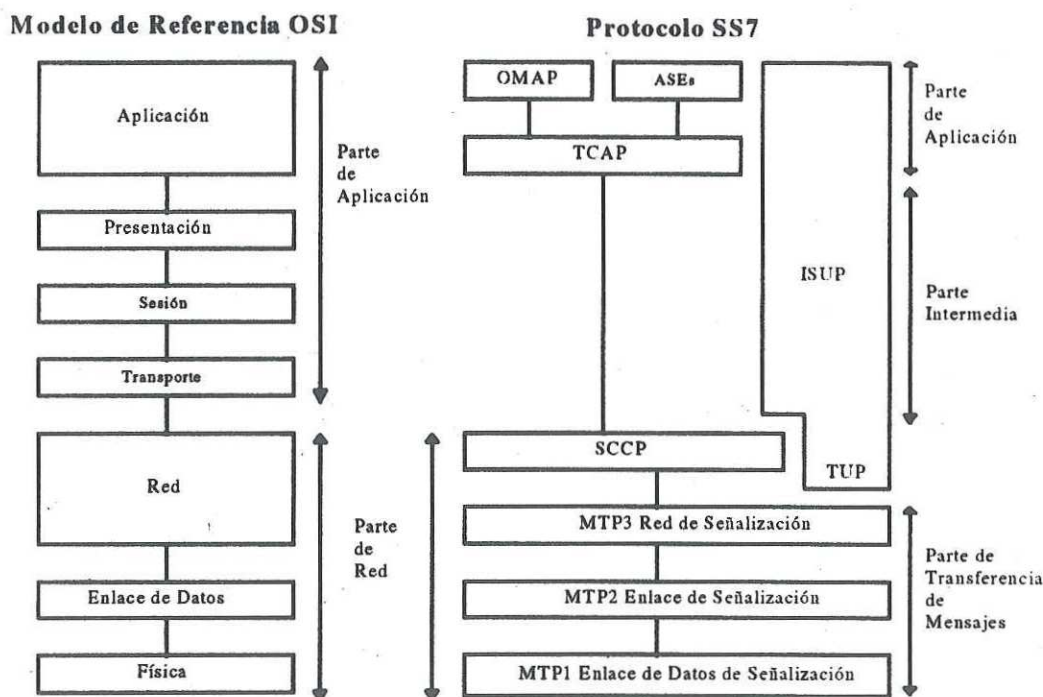


Figura 8 Comparación del modelo de referencia OSI con el protocolo SS7.

El protocolo está dividido en diferentes partes, consistiendo en varios niveles (ver figura 8). Estas partes son definidas en términos de la red o la funcionalidad de servicios, los cuales se encuentran reflejados dentro de las principales entidades físicas que conforman la red inteligente tales como el SSP, STP y SCP, como se muestra en figura 9, estas son (Thorner, 1994):

- Parte de Transferencia de Mensajes

- Parte de Red
- Parte Intermedia
- Parte de Usuario
- Parte de Aplicación

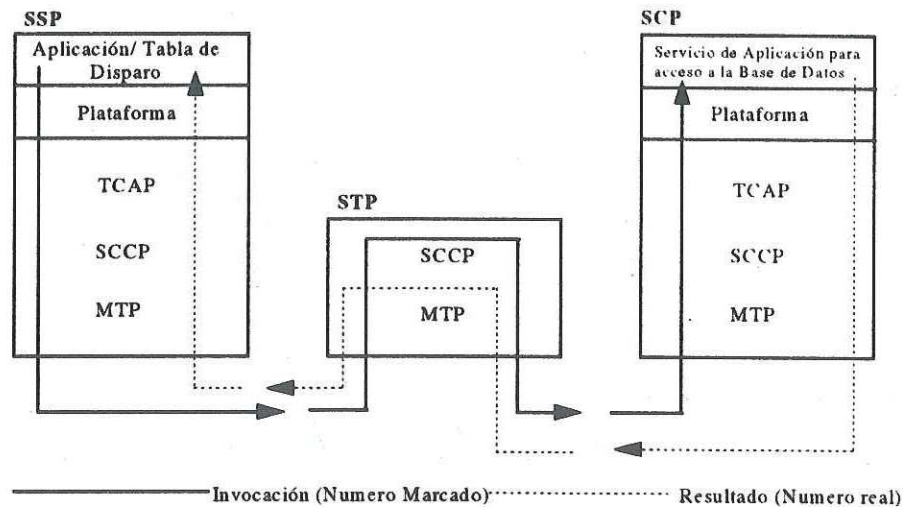


Figura 9 Relación del Protocolo SS7 en las entidades físicas de Red Inteligente

La capa física del modelo de referencia OSI, la capa de enlace de datos y algunas partes de la capa de red, se encuentran en la parte de transferencia de mensajes y por lo tanto, los niveles de MTP 1-3 son responsables de las funciones de confiabilidad de la red en SS7. El MTP consiste de tres niveles en el protocolo; MTP1, MTP2 y MTP3.

Los niveles MTP3 y SCCP forman parte de la capa de red en el modelo de referencia OSI, la cual ofrece servicio tales como el establecimiento de conexión lógica, manejo de conexión y la capacidad de la transferencia de datos secuenciados, los cuales son básicamente funciones relacionadas a la capa de transporte de modelo de referencia OSI.

Las recomendaciones no definen las capas del modelo de referencia OSI 4 a la 6 para el protocolo SS7. Sin embargo, las funciones de estas capas están especificadas en otro nivel del protocolo SS7, como en el SCCP. Los niveles intermediarios son colectivamente llamadas parte intermedia.

El protocolo SS7 puede incluir una parte de usuario, la cual define funciones básicas para el control de la llamada con un protocolo TUP (Parte de Usuario, Transaction User Part) o llamadas ISDN con un protocolo ISUP (Parte de usuario ISDN, ISDN User Part). La parte de aplicación define las interfaces para funciones específicas tales como consultas a bases de datos y servicios de Red Inteligente .

II.4.1 PARTE DE TRASFERENCIA DE MENSAJES (MTP)

La MTP provee un servicio confiable en el enrutamiento de información a través de la red SS7. Como ya mencionamos anteriormente esta parte se divide en tres niveles en donde MTP1 maneja las operaciones de enlace de datos de señalización SS7 y define las características de la capa física del modelo OSI en la red SS7. MTP2 provee las funciones de enlace de señalización y es el equivalente a la capa de enlace de datos del modelo OSI, en donde se establece la entrega confiable de paquetes de datos secuenciados a través de MTP1.

El enrutamiento entre diferente puntos de señalización en la red SS7 está especificado en MTP3. En donde se definen las direcciones de red usando un código (PC, Point Code), en donde cada código direcciona a un punto de señalización único en la red SS7. El código origen (OPC, Originating Point Code) identifica el punto de señalización

fuelle que envi6 el mensaje y el c6digo destino (DPC, Destination Point Code) identifica al punto de se1alizacion destino que recibir6 el mensaje. Por lo tanto, MTP3 es responsable del manejo de la se1alizacion. La figura 10 nos muestra la estructura general de la MTP en la red SS7.

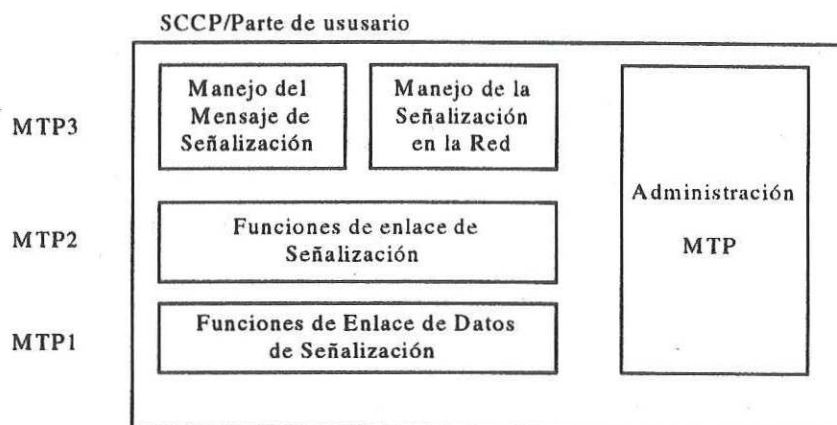


Figura 10 Estructura general del MTP en la red SS7

II.4.2 PARTE DEL CONTROL DE LA CONEXIÓN DE SE1ALIZACIÓN (SCCP)

La SCCP provee los medios para:

- Control de se1alizacion en conexiones l6gicas en la red SS7.
- Transferencia de unidades de datos de se1alizacion en la red SS7.

Tambi6n la SCCP, define las capacidades de direccionamiento en la red, las cuales pueden ser divididas en las siguientes categorías:

- Títulos Globales (GT)
- DPC y Numero de subsistema (SSN, Subsystem Number)

GT es una direcci6n como un conjunto de n6meros, que no contienen informaci6n

explícita de enrutamiento pero pueden ser usados para identificar al abonado que llama y al abonado al que se le llama. El enrutamiento del mensaje entre puntos de señalización es hecho usando DPC y la indicación del servicio requerido usando SSN. En donde DPC define la dirección de red del punto de señalización (por ejemplo, la dirección del SCP), por otra parte el SSN identifica la dirección lógica de la parte de aplicación. La SCCP por otra parte ofrece dos clases de servicio para el usuario o para la parte de aplicación, éstos son: servicios orientados a conexión o servicios orientados a no conexión, como se muestra en la figura 11.

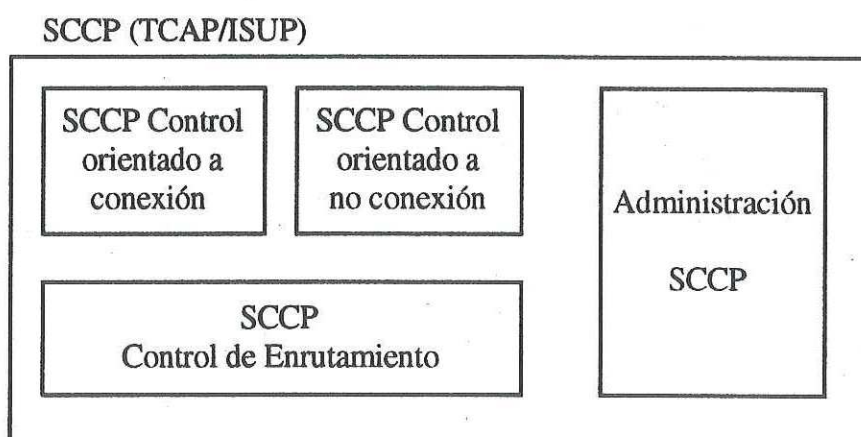


Figura 11 Estructura general del SCCP en la red SS7.

II.4.3 PARTE DE APLICACIÓN DE LAS CAPACIDADES DE TRANSACCIÓN (TCAP, Transaction Capabilities Application Part)

La TCAP en la red SS7 consiste de dos sub-partes:

- TCAP
- TCAP de usuario (Elemento de servicios de aplicación, ASE, Application Service

Element) tal como Protocolo de aplicación de Red Inteligente (INAP, IN Application Protocol) (Q.1208, 1993).

En términos del modelo OSI, TCAP es un ASE, el cual es responsable del manejo del dialogo entre los puntos de señalización distribuidos en la red SS7. Éste consiste en dos subcapas llamadas "Porción de componentes" y "Porción de la Transacción" como se muestra en la figura 12.

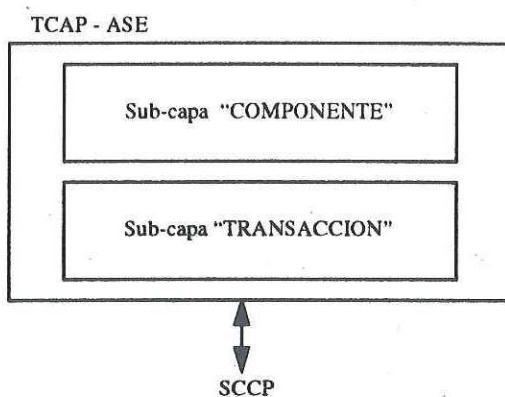


Figura 12 Estructura de TCAP - ASE.

La sub-capa de componente maneja las funciones de llamadas remotas para los servicios, también codifica y decodifica las parámetros de las llamadas. La sub-capa de transacción maneja el dialogo de usuario entre los puntos de señalización, el intercambio de unidades de datos de protocolos o PDUs (Protocol Data Unit) (Q.1208, 1993).

TCAP provee procedimientos para soportar una variedad de aplicaciones, evitando así la ineficiencia de procedimientos específicamente ajustados. La TCAP provee un marco de trabajo para un enfoque común para servicios nuevos dentro de una red y para la arquitectura de servicios de inter-red. La TCAP usa reglas estándares para su formato, por

lo tanto el procedimiento de su formato de trama (identificador, longitud y valor) sirve para todos los usos. Además, con la transacción de subcapas, los usuarios pueden intercambiar información en un camino estructurado o no estructurado. Actualmente, solo el usuario de la subcapa de transacción puede usar la aplicación para invocar operaciones y reportar sus resultados intercambiando componentes. Los componentes definidos son:

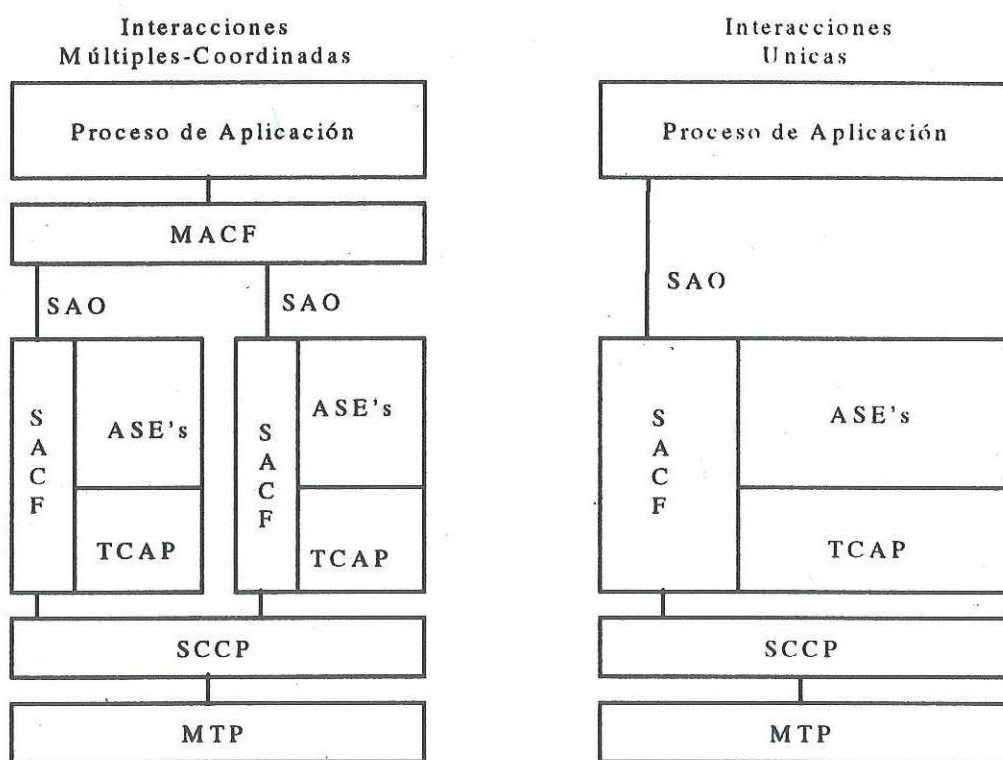
- Invocar: Usado para llevar a cabo una operación.
- Resultado de Regreso (RR): Reporta un suceso y regresa el resultado de una operación invocada.
- Regreso de Error (RE): Reporta la recepción de un mensaje causando un error de protocolo en la transacción.

II.4.4 PARTE DE APLICACIÓN DE RED INTELIGENTE (INAP, IN Application Part)

INAP se define como una colección de ASEs, los cuales son definidos como un conjunto de funciones de aplicación que proveen la capacidad de trabajar con las entidades de aplicación para propósitos específicos (Q.1208, 1993).

INAP corresponde a la subcapa de “componente” de TCAP, la cual es utilizada para la comunicación en la Red Inteligente, por ejemplo entre el SSP y el SCP. La arquitectura del protocolo INAP se muestra en la figura 13. Por lo tanto, con INAP, los puntos de señalización pueden tener relaciones múltiples o únicas con otros puntos de señalización. En otras palabras, una entidad física puede tener interacciones únicas o múltiples combinadas con otras entidades físicas.

En el caso de interacciones únicas, SACF (Función de control de asociación única) provee una función de coordinación usando ASEs con el fin de soportar el ordenamiento de operaciones (basado en la recepción de primitivas). La asociación de Objetos únicos conocidos por SAO representan al SACF mas un conjunto de ASEs que son usados sobre una sola interacción entre un par de entidades físicas. En el caso de múltiples interacciones (MACF) se provee una función de coordinación entre varios SAOs.



SACF Función de Control de Asociación Unica
 MACF Función de Control de Asociación Múltiple
 SAO Objeto de Asociación Unica
 ASE Elemento de Servicio de Aplicación
 INAP Protocolo de Aplicación de Red Inteligente

Figura 13 Arquitectura del Protocolo INAP.

II.5 CONJUNTO DE CAPACIDADES DE LA ITU-T PARA RED INTELIGENTE

La ITU es una agencia especializada de las Naciones Unidas (UN) dentro de la cual el gobierno y el sector privado de telecomunicaciones coordinan el establecimiento y operación de redes y servicios de telecomunicaciones. La ITU-T es responsable de regular, estandarizar, coordinar y desarrollar todo lo referente a las telecomunicaciones internacionalmente. En donde la ITU-T lleva a cabo conferencias para revisar y modificar lo relacionado a sus tres sectores: ITU-R Sector de Radiocomunicación; ITU-D Sector de Desarrollo; ITU-T Sector de Telecomunicaciones. La tarea principal de la ITU-T es el estudio técnico, operativo y cuestiones relacionadas a la tarificación, y edición de recomendaciones (estándares) con el fin de estandarizar las telecomunicaciones en bases globales. En donde las recomendaciones cubren aspectos internacionales de telecomunicaciones incluyendo la interconexión de sistemas de radio en redes públicas de telecomunicaciones (Robart, 1997).

Los objetivos de la Red Inteligente, definidos por la ITU-T, se listan a continuación:

- El uso extensivo de técnicas de procesamiento de información y recursos de red.
- La integración de la creación e implementación de servicios usando funciones de red estándares y reutilizables.
- Asignación flexible de funciones reutilizables de red a través de las entidades físicas.
- Comunicación estandarizada entre funciones de red, vía interfaces independientes.
- Permitir el acceso a proveedores de servicio para procesar servicios a través de la combinación de funciones de red.

- Permitir a los subscriptores de servicios controlar la definición y despliegue de atributos en servicios específicos del subscriptor.
- Permitir el manejo de lógica de servicio estandarizada.

La realización de estas características pueden estar definidas usando un método de intercambio a través de las recomendaciones que forman parte del conjunto de capacidades (CS, Capabilities Set). Los CSs asimismo direccionan a un incremento de funciones, alcanzables dentro de cierto plazo de tiempo. Tal y como se muestra en la figura 14. Este proceso representa la evolución de las características y capacidades en la Red Inteligente definida por la ITU-T.

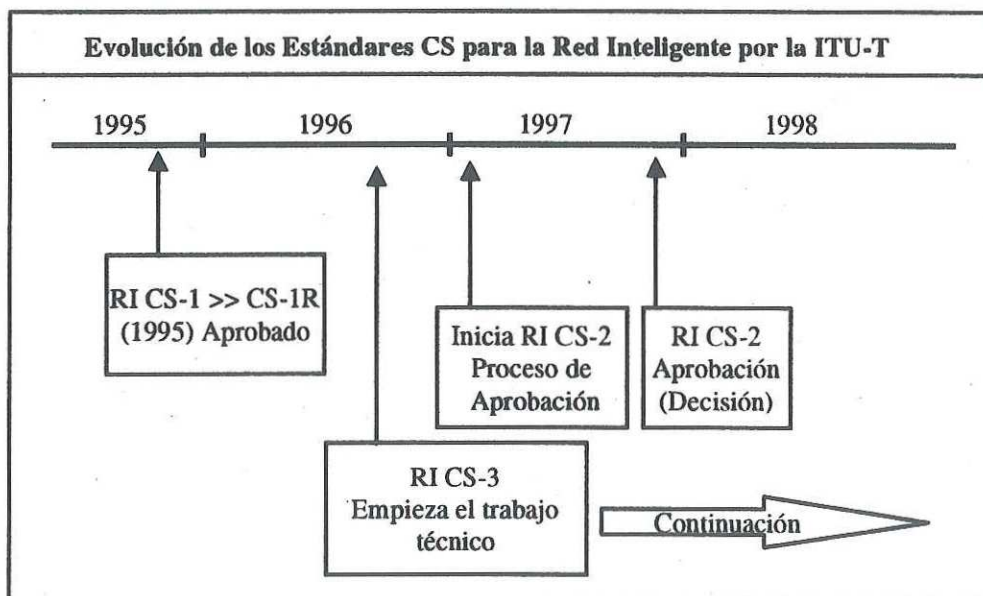


Figura 14 ITU-T RI-CS Estandarización.

La ITU-T en su recomendación Q.1200 define la estructura general de las recomendaciones para la RI como:

Q.12x1 : Introducción

Q.12x2 : Plano de Servicio

Q.12x3 : Plano Global Funcional

Q.12x4 : Plano Funcional Distribuido

Q.12x5 : Plano Físico

Q.12x8 : Recomendación de Interfaz INAP

Q.12x9 : Guía de Usuario

Q.1290 : Vocabulario

En donde Q.121x, Q.122x y Q.123x representan el conjunto de recomendaciones de RI CS-1, RI CS-2 y RI CS-3 respectivamente. En la figura 15, podemos ver la estructura de las recomendaciones para la RI por la ITU-T (Unger, 1997).

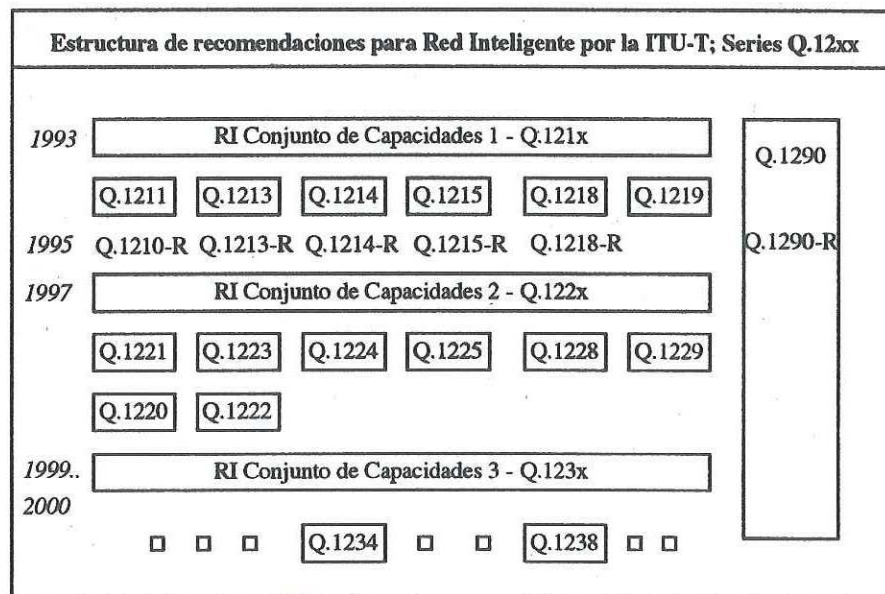


Figura 15 ITU-T estructura de recomendaciones para la RI.

La recomendación CS-1 para la Red Inteligente fue producida en 1993. Mejoras a estas recomendaciones fueron hechas en el transcurso de 1993-1995 resultando CS-1R versión mejorada, la cual fue producida en 1995. CS-1R provee una definición inicial de Red Inteligente. Ésta define las capacidades de soportar un solo punto terminal, un solo punto de control de servicio, proveyendo servicios tales como: (Duran y Visser, 1992)

- Enrutamiento Distribuido de Llamadas (CRD, Call Rerouting Distribution) : Permite al subscriber enrutar una llamada hacia diferentes destinos usando una distribución circular, por porcentaje o jerárquica.
- Comunicación Tripartita (CON, Conference Call) : Permite a múltiples usuarios estar conectados en una sola conversación.
- Llamada con tarjeta de crédito (CCC, Credit Card Calling) : Permite al subscriber que el costo de la llamada sea cargado automáticamente a su tarjeta de crédito. El subscriber tendrá que introducir su número de tarjeta de crédito y un número de acceso (PIN) vía tonos multi-frecuencias ó a través de la operadora.
- Enrutamiento de llamada a un destino específico (DCR, Destination Call Routing) : El usuario podrá enrutar su llamada a un destino específico, dependiendo del día, hora, fecha, área de origen de la llamada, etc.
- Seguimiento al destinatario (FMD, Follow-Me Diversion) : El usuario podrá enrutar su llamada a una sola dirección.
- Teléfono libre de cobro (FPH, FreePhone) : Permite al usuario llamar gratuitamente a un destino, cargando el costo de la llamada hacia el destino.

- Identificación de llamadas maliciosas (MCI, Malicious Call Identification) : Soporta la identificación de llamadas que llegan de localidades específicas.
- Llamadas masivas (MAS, Mass Calling) : El operador de la red puede asignar temporalmente un solo número del directorio para servicios de usuario.
- Televoto (VOT, Televote) : Similar al anterior, en donde se permite que el usuario use un solo número para votar dentro de un evento televisado.
- Número personal universal (UPT, Universal Personal Telecommunication) : UPTs es un servicio móvil que permite al usuario hacer uso del servicio con un solo número personal a través de la red y a través del acceso de varias redes.
- Redes virtuales privadas (VPN, Virtual Private network) : Permite a los clientes construir una red privada utilizando los recursos de la red pública.

RI CS-1R define también la interfaz entre las funciones SSF/CCF con la SCF. Esta interfaz típicamente comunica el conmutador y las bases de datos de la Red Inteligente (SCP). Operaciones entre el SCF hacia el SRF, típicamente relaciona operaciones entre las bases de datos (SCP) y un IP.

El desarrollo de RI CS-2 empezó en 1995 y fue recientemente propuesto por el grupo de estudio número 11 de la ITU-T en Enero de 1997. En donde la aprobación final está planeada para septiembre de 1997. RI CS-2 contiene todo acerca de CS-1R (1995) e incluye capacidades para soportar servicios como: Manejo de llamada (CPH, Call Party Handling), servicios inalámbricos, servicios de seguridad y servicios no asociados a llamadas. Además de lo propuesto en la mejora de CS-1, en CS-2 se definen interfaces de

inter-red para:

- SDF a SDF, interface física que provee la distribución de datos.
- SCF a SCF, típicamente de SCP a SCP, esta interfaz provee la capacidad de distribución de la lógica de servicio.
- SMF a SMF, típicamente de SMS a SMS.

El estudio técnico en CS-3 empezó en 1996. CS-3 en general soporta todos los aspectos evolutivos de CS-2, movilidad y la integración de la RI de banda ancha (RI/Broadband). En Enero de 1997 el grupo de estudio número 11 de la ITU-T acordó que la recomendación CS-3 será dividida en dos versiones: RI CS-3.1 una versión a corto plazo y RI CS-3.2 una versión a mediano y largo plazo.

Las principales características de CS-2 y CS-3 serán enfocadas a las siguientes capacidades de servicio:

- Soporte de llamadas Multimedia.
- Videoconferencia.
- Servicios de Banda Amplia.
- Estandarización en la creación, despliegue y manejo de servicios.
- Representación de datos.
- Movilidad.
- Interoperabilidad de la RI con otras redes.

III. ESTUDIO DE MECANISMOS DE CONTROL

III.1 INTRODUCCIÓN

Los esquemas de control de flujo y congestión son muy importantes para el diseño y administración de redes de telecomunicaciones. Si la demanda sobrepasa la capacidad del sistema ya sea en redes telefónicas o en redes de conmutación de paquetes, el sistema pasará a un estado de congestión. Esto causará que los elementos del sistema sean pobremente utilizados, o que los tiempos de respuestas se incrementen y el caudal eficaz en el sistema decrecerá rápidamente. Por lo tanto, para resolver esta situación, la red debe estar provista de mecanismos de control que puedan mantener el flujo de datos dentro de los límites compatibles a la cantidad de recursos disponibles en el sistema.

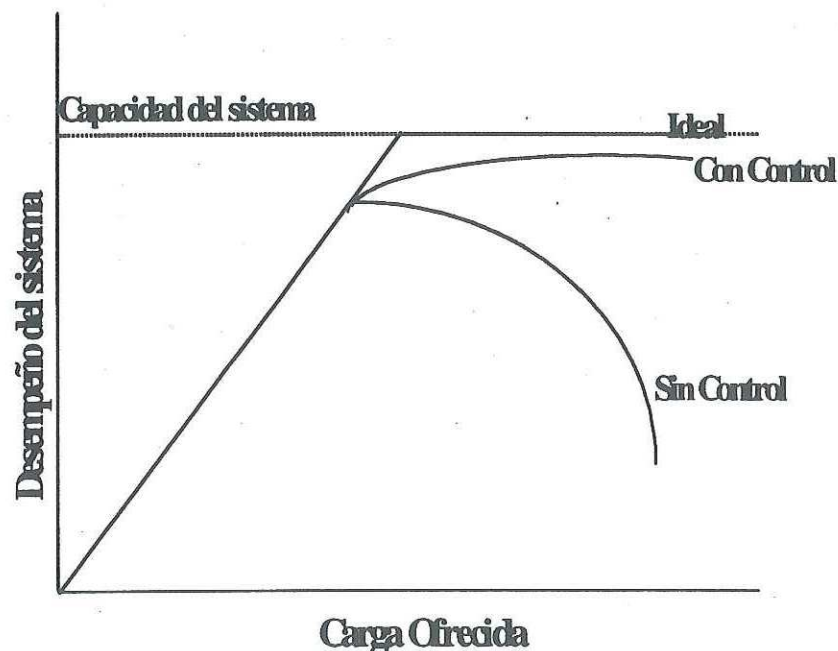


Figura 16 *Desempeño del sistema Vs Carga ofrecida.*

La figura 16, nos muestra como el comportamiento de un sistema se degrada si la cantidad de trabajo sobrepasa la capacidad del sistema. La curva inferior nos muestra el decremento dramático del desempeño del sistema debido a congestión (*sin control*). La curva superior nos muestra el comportamiento *ideal o perfecto* del sistema. La curva del medio representa el comportamiento del sistema alcanzado gracias a la aplicación de un mecanismo de control efectivo (*con control*). Los mecanismos de control de flujo y congestión son necesarios para prevenir un gran decremento en el desempeño del sistema, y tratar de llegar tan cerca como sea posible al comportamiento ideal o perfecto. En la práctica el tráfico o la carga ofrecida puede que sea completamente desconocida o difícilmente predecible. Entonces, se necesita un mecanismo de control que sea robusto, es decir, que la arquitectura del mecanismo sea relativamente insensible a cualquier cambio.

III.2 DEFINICIÓN DE MECANISMO DE CONTROL DE FLUJO

- Control de flujo es un conjunto de mecanismos con los cuales un flujo de paquetes puede ser mantenido dentro de los límites compatibles a la cantidad de recursos disponibles (Pouzin, 1981).
- Control de flujo es un acuerdo entre una fuente y un destino para limitar el flujo de paquetes sin tomar en cuenta la carga en la red (Jain, 1986).

De estas definiciones, podemos concluir que control de flujo es un método el cual asegura que un paquete que llega a su destino encontrará recursos disponibles. Por lo tanto, los mecanismos de control de flujo limitan el tráfico basado en la disponibilidad del destino.

Un mecanismo de control de flujo lleva a cabo un conjunto de funciones tales como:

- Prevenir la pérdida de tráfico que podría ocurrir cuando un receptor está sobrecargado,
- evitar interferencias entre usuarios,
- minimizar el tiempo de respuesta en la comunicación,
- maximizar el caudal eficaz,
- minimizar el costo de la transmisión,
- mantener el receptor ocupado todo el tiempo,
- minimizar la posibilidad de sobre flujos.

Los esquemas de control de flujo suelen ser una combinación de varias funciones trabajando en diversas capas en una arquitectura de protocolos. En práctica, cualquier implementación de mecanismos de control de flujo involucran un número limitado de herramientas para reducir el tráfico como son:

- Para y sigue (Stop and Go): En donde la fuente es controlada por una señal de naturaleza binaria. En donde ésta puede enviar tráfico sin límite, o con límite. Esta técnica es usada en algunos procedimientos de control de enlace de datos, tal como en la técnica de control de enlace de datos de alto nivel (HDLC, High-level Data Link Control).
- Crédito (credit): La fuente no puede transmitir a menos que reciba del receptor una indicación acerca de la cantidad de tráfico que puede aceptar. Este tipo de esquemas protegen completamente al receptor de un exceso de llegadas de paquetes.
- Tasa de llegadas (Rate): En donde la fuente transmite tráfico a una determinada tasa, la

cual puede haber sido evaluada de acuerdo a los recursos requeridos. La tasa puede ser ajustada dinámicamente cuando los recursos cambian.

- Clase (Class): En donde el flujo de tráfico pertenece a diferentes clases o tipos. Cuando suficientes recursos están disponibles, todo tráfico es aceptado. De otro modo, algunas clases son rechazadas.

Los esquemas de control de flujo son una mezcla de dos distintos componentes, los cuales deben ser claramente distinguidos.

Protocolo de control : El primer componente es un protocolo de control de flujo, por ejemplo, un conjunto de reglas seguidas por el transmisor y el receptor para intercambiar información de control concernientes al flujo de datos. Por ejemplo: señales, créditos, tasas, prioridades, etc.

Administración de recursos : El segundo componente es un esquema de administración de recursos, por ejemplo, un mecanismo que mantenga la información de todos los recursos que estén asignados y que se puedan asignar para llevar a cabo un proceso.

III.3 DEFINICIÓN DE MECANISMO DE CONTROL DE CONGESTIÓN

- Control de congestión es un conjunto de mecanismos según los cuales, un subsistema intermediario mantiene la entrada de tráfico dentro de los límites compatibles con la cantidad de recursos disponibles en el subsistema. (Pouzin, 1981).
- Control de congestión se refiere principalmente al control de tráfico para reducir el sobreflujo en la red (Jain, 1986).

De estas definiciones podemos concluir que los mecanismos de control de congestión son aquellos capaces de llevar a cabo una estrategia para reducir el impacto del sobreflujo en el sistema, es decir, son aquellos que limitan a las fuentes en los periodos que el sistema no puede cumplir con toda la carga ofrecida.

Las principales estrategias de control de congestión son:

- 1) preasignar recursos para evitar la congestión,
- 2) disponer de nodos intermediarios para descartar tráfico y enviarlos hacia ellos,
- 3) restringir el número de paquetes permitidos,
- 4) usar un mecanismo de control de flujo, y
- 5) limitar la entrada de tráfico en periodos de congestión.

Existen varios métodos para llevar a cabo el control de congestión pero los más usados son aquellos capaces de disminuir el tráfico de entrada sólo con la observación. La mayoría de los estudios se han centrado en la administración de memoria en los conmutadores. Para la administración de memoria, se han estudiado varios como los llamados Isaritmicos, limitar la entrada del Buffer (Input Buffer Limit), y clases de Bufers (Buffer Class).

- Isaritmico: La principal idea de este esquema es mantener el número total de paquetes en tránsito bajo un cierto umbral, el cual puede ser considerado como la máxima carga en la red.
- Limitar la entrada de paquetes a la memoria: En donde cada nodo es capaz de almacenar una cierta cantidad máxima de paquetes. En donde los paquetes que están por llegar a su destino tienen una mayor prioridad. Así como los paquetes en tránsito tienen una

segunda prioridad, mientras que los paquetes que recién entran a la red tienen la menor prioridad y no pueden tomar más que un cierto número en el buffer.

- Clase de Buffer: En donde los buffers son divididos en diferentes clases como el número de saltos a lo largo de la ruta a través de la red. Donde se espera para el siguiente nodo, un paquete que sólo puede ser recibido dentro de un buffer según la clase correspondiente al número de saltos que el paquete ha dado.

III.4 DIFERENCIA ENTRE CONTROL DE FLUJO Y CONTROL DE CONGESTIÓN.

Algunos autores consideran el control de congestión como un caso especial de control de flujo (Tanenbaum, 1997). Ambos control de congestión y de flujo involucran la reducción de velocidad con la cual las fuentes transmiten tráfico. El control de flujo previene que la fuente congestione al destino, mientras que el control de congestión mantiene la sobrecarga de la red. Ambos necesitan ser capaces de informar a la fuente, la cantidad de paquetes que puede transmitir. Existen dos formas informarle a la fuente: con el uso de una ventana o con una tasa de llegadas. La ventana corresponde a que tanto espacio de memoria disponible tiene el receptor y esto a su vez limita la cantidad de paquetes que la fuente puede transmitir; esto es soportado por el control de flujo. Esta misma información puede ser usada dentro de la red para reservar el espacio del buffer, esto es soportado por el control de congestión. Por otra parte, también es posible expresar el control de flujo y congestión en términos de tasa de llegadas, es decir, por ejemplo cuantos paquetes por segundo el receptor o la red puede procesar.

Podemos concluir que el control de flujo trata de evitar que la fuente congestione al receptor, además, el control de flujo es un mecanismo de común de acuerdo entre dos puntos o entidades.

Por otra parte, el control de congestión trata de evitar que la red esté sobrecargada. Además control de congestión se le conoce como una ley social, es decir, considera toda la red en general.

III.5 DESCRIPCIÓN DE MECANISMOS DE CONTROL EN LA RED INTELIGENTE

III.5.1 CONTROL DE BANDAS DE LLAMADAS (CALL GAPPING)

El primer esquema de control que estudiaremos es el llamado Call Gapping (Pham y Betts, 1994). En (Fukuda, 1986) Call Gapping es considerado como un mecanismo de control de entrada basado en monitoreos periódicos en un modelo de colas finitas. Call Gapping es activado y desactivado comparando el número de llamadas en el sistema con dos valores de umbrales. En nuestro estudio, este mecanismo de control es implementado en el SCP, donde el SCP debe de llevar a cabo periódicamente estadísticas detalladas del número de llamadas procesadas y enviadas como respuestas a peticiones de servicios al SSP, cuando el número de llamadas enviadas por los SSPs y almacenadas en la memoria del SCP al inicio de cada período excede un umbral superior, el cual nos representa un posible estado de congestión, el mecanismo de Call Gapping es activado, ajustando la tasa de llegadas de todos los SSPs. Para ello se envía un mensaje de control en el cual se especifica el valor de

la banda (g) la cual se agrega al tiempo de servicio del SSP, gobernando así el tráfico de Poisson a la entrada, donde “ g ” actúa como un modulador. El valor de la banda es determinado cumpliendo con la siguiente relación básica (Smith, 1995) :

$$T_{out} = g + T_{in} \quad (1)$$

En donde T_{out} es el tiempo medio entre salidas de llamadas del SCP en el período de monitoreo, T_{in} es el tiempo medio estimado entre entradas de llamadas al SCP en el período de monitoreo y g es el tiempo entre llegadas sobre impuesto al tiempo de servicio del SSP. Por lo tanto si el número de llamadas sobrepasa el umbral superior el valor de g se calcula como sigue:

1. Permitamos que b sea un factor de reducción (igual a 0.9 cuando el número de llamadas sobrepasa el umbral superior, el cual corresponde a un 10% en la reducción de llamadas).
2. El SCP calcula el T_{out} y usa el valor actual de g (inicialmente igual a 0) para estimar T_{in} usando la ecuación (1).
3. El SCP incrementa T_{out} por un factor de $1/b$ y calcula g usando la ecuación (1).

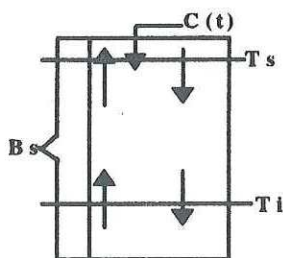
Una vez que el SCP ha calculado g , este envía un mensaje de control a los SSPs indicando el nuevo valor de g . Del mismo modo, la banda previa g es cancelada cuando el número de llamadas en el SCP disminuye por debajo del valor del umbral inferior, enviando un mensaje de control a todos los SSPs con un nuevo valor de g implícito. El nuevo valor de g se calcula usando la ecuación (1) por el SCP después de multiplicar a T_{out} por b . Este procedimiento decrementa el valor de g .

La figura 17, nos muestra el esquema y el algoritmo del comportamiento e implementación del mecanismo Call Gapping respectivamente. Sea T_i el umbral inferior y T_s el umbral superior, $C(t)$ el número de llamadas en el buffer del SCP en tiempo t , y B_s el tamaño del buffer en el SCP. El mecanismo Call Gapping es manejado cumpliendo las siguientes desigualdades:

Si $0 \leq C(t) \leq T_i - 1$; El mecanismo de control no es activado o si el mecanismo de control fue previamente activado se desactiva calculando un nuevo valor de g .

Si $T_i \leq C(t) \leq T_s$; El mecanismo de control es mantenido en el mismo estado.

Si $T_s + 1 \leq C(t) \leq B_s$; El mecanismo de control es activado y el SCP envía el valor de la banda g impuesta a los SSPs.



```

If (Tiempo muestreo) then
If  $0 \leq C(t) \leq T_i - 1$  then
  If (activar) then
    :desactivar
  else
    :continuar
If  $T_i \leq C(t) \leq T_s$  then
  :continuar
If  $T_s + 1 \leq C(t) \leq B_s$  then
  :Call Gapping
else
  :continuar

:continuar
  mantiene el mismo estado
:Call Gapping
  activar = si
  Calcula nuevo valor de  $g$ 
  Envía valor de  $g$  al SSP
:desactivar
  activar = no
  Envía nuevo valor de  $g$ 
  normalizado al SSP
  
```

Figura 17 Mecanismo de Control Call Gapping.

En redes que utilizan el mecanismo de control de congestión Call Gapping, el principal parámetro que determina la carga en el buffer del nodo destino y de ahí el desempeño del mismo, es el valor de la banda g calculada por el mecanismo cada vez que el nodo destino monitorea un posible estado de congestión, comparando la cantidad de tráfico con dos umbrales como se explicó anteriormente. Este mecanismo hace uso de dos especificaciones: tiempo periódico de muestreo y los valores de los umbrales. Un número de alternativas fueron escogidas para seleccionar los valores óptimos para llevar a cabo este mecanismo, en el capítulo IV veremos como se seleccionaron por medio de simulación estos valores. Sin embargo a continuación explicaremos qué se debe de tomar en cuenta para la selección de estos valores.

- 1) Tiempo periódico de muestreo: Se deberá seleccionar un tiempo de muestreo capaz de minimizar la probabilidad de bloqueo y a la vez optimar el tiempo de procesamiento para llevar a cabo el mecanismo de control.
- 2) Valor de los umbrales: Call Gapping hace uso de dos umbrales (superior/inferior) para monitorear un posible estado de congestión. La relación entre los dos umbrales deberá ser la óptima, capaz de permitir que el mecanismo no tenga muchas fluctuaciones y que el tiempo de respuesta por la fuente para implementar el control administrado por el destino no sea muy largo. Por otra parte, buscar minimizar la probabilidad de bloqueo en el sistema.

III.5.2 CONTROL DE LA VENTANA (WINDOW)

La red inteligente necesita un mecanismo de control que mantenga informada a la fuente cuando la red está en congestión y un mecanismo para que la fuente ajuste su carga en la red. Por lo tanto, el mecanismo de control de la ventana proporciona una manera simple de ajustar la carga en la fuente (Pham y Betts, 1994), (Peterson y Bruce, 1996). Una red existente que utiliza el control de flujo de la ventana es ARPAnet.

En este mecanismo, la fuente utiliza un valor de ventana que corresponde al tamaño del buffer del receptor, siendo esto lo que limita a la fuente en la cantidad de paquetes que puede enviar. En este mecanismo, el SCP no usa ningún mensaje de control especial hacia los SSPs. En nuestro estudio, el mecanismo de control de la Ventana es implementado en los SSPs, donde dos variables son usadas para determinar cuando una llamada puede ser enviada hacia el SCP, la primera de ellas es llamada *WIN* la cual indica el número de llamadas permitidas para ser enviadas en cualquier tiempo, y la segunda variables es un contador llamado *OUT* usado para conocer el número de llamadas que fueron enviadas hacia el SCP.

La figura 18, nos muestra un diagrama de flujo, donde podemos ver como hemos implementado este mecanismo en el SSP, donde si $OUT=WIN$, la ventana de transmisión se cierra, en este momento el SSP no puede seguir enviando llamadas hacia el SCP y este empieza a almacenarlas. Cuando $OUT \neq WIN$ el SSP puede seguir mandando más llamadas hacia el SCP. Por otra parte cuando una respuesta es recibida por el SSP este incrementa su variable *WIN* solo si $WIN < WinMax$, donde *WinMax* es el valor máximo que la ventana

puede tomar. Cuando una llamada es enviada al SCP, se activa un temporizador. Si el temporizador expira nos permite decir que el SCP está en congestión o la llamada se ha perdido por fallas en el sistema, en este momento la variable *WIN* es decrementada. Del mismo modo al momento de recibir una respuesta el SSP, compara el tiempo en que se envió la llamada con el tiempo en la que llegó, si este valor es menor que t_s la llamada ha sido servida exitosamente, sino de la misma manera, la variable *WIN* se decrementa. ($t_s = 2s$, tiempo máximo en segundos permitidos por una llamada inteligente a ser cumplida).

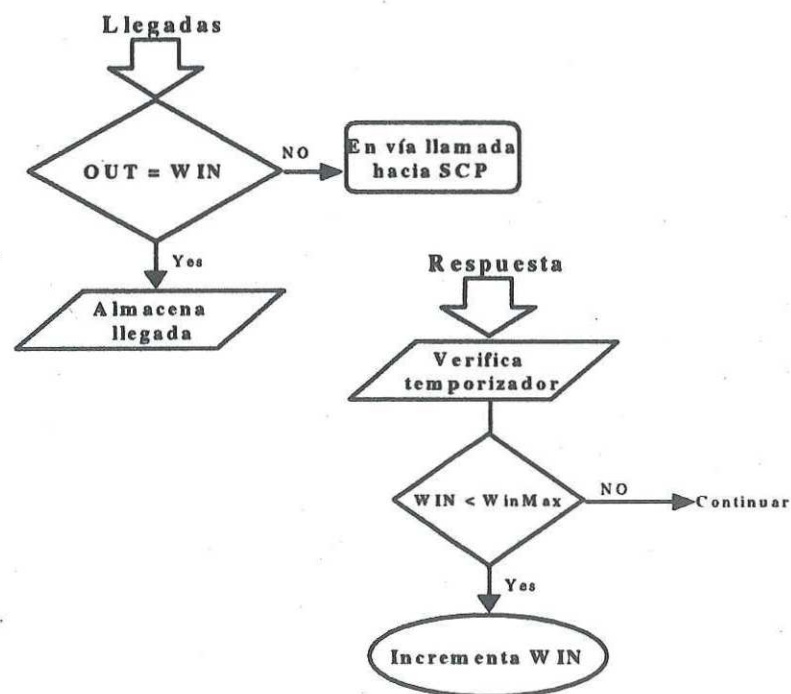


Figura 18 Mecanismo de Control de la Ventana.

En redes que utilizan el mecanismo de control de flujo de la ventana, el principal parámetro que determina la carga en la red y de ahí su desempeño es el tamaño de la ventana usado por las fuentes. Es claro que si el tamaño de la ventana se incrementa, el

caudal en la red se verá incrementado. Sin embargo, si el tamaño de la ventana es muy grande y sobrepasa la capacidad de tamaño de los buffers en la red, el caudal empezará a decaer perdiéndose paquetes o llamadas en la red. Así, el tamaño de la ventana en momentos de congestión es conocido como la capacidad de los buffers. En el caso de múltiples fuentes (SSPs en el caso de red inteligente), la suma del tamaño de la ventana de cada uno de ellos, deberá ser menor o igual a la capacidad del buffer destino (el SCP en el caso de red inteligente).

El mecanismo de la ventana hace uso de cinco elementos: Un valor de ventana máximo, un valor de ventana mínimo, una política de inicialización de la ventana, una política de incremento y una política de decremento del valor de la ventana. Un número de alternativas fueron escogidas para seleccionar los valores óptimos para llevar a cabo este mecanismo, en el capítulo IV veremos como se seleccionaron por medio de simulación estos valores. Sin embargo, explicaremos a continuación la utilización de cada uno de ellos.

- 1) Máximo: El tamaño de la ventana máxima (WinMax) nunca deberá ser mayor a la capacidad del buffer del receptor (SCP).
- 2) Mínimo: El tamaño mínimo de la ventana (WIN) es igual a uno, así en momentos en los que se reduce la ventana por congestión o fallas en la red, el tamaño de la ventana nunca podrá ser menor a este valor.
- 3) Inicialización: El tamaño inicial de la ventana no tiene mucho efecto en el desempeño de la red en un período largo. Sin embargo, en una red muy poco cargada es más eficiente empezar con WinMax, en cambio en una red muy cargada es mas seguro empezar con el valor mínimo.

- 4) Incremento: El tamaño de la ventana (WIN) puede ser incrementado de uno en uno después de un número de paquetes o llamadas servidas hasta llegar al valor máximo permitido. Sin embargo, se considera eficiente si el tamaño de la ventana incrementa en uno por cada paquete o llamada servida por el destino.
- 5) Decremento: Entre las cinco especificaciones, está llega a ser la más importante en términos de desempeño en la red. Ésta puede ser como:
- Decremento Repentino: El tamaño de la ventana es reducido al valor mínimo.
 - Decremento Gradual: El tamaño de la ventana se reduce en uno cada vez que un temporizador es activado.
 - Decremento Binario: El tamaño de la ventana se reduce a la mitad de su valor y se redondea a un entero.

III.5.3 CONTROL A DOS NIVELES (TWO LEVELS CONTROL)

Ventana y Control de Sobre Flujo.

En este trabajo de tesis, proponemos un nuevo mecanismo de control a dos niveles:

1) un mecanismo de control de flujo es usado como regulador del número de llamadas enviadas hacia el SCP por unidad de tiempo y, 2) un control de sobre flujo es implementado en el SCP para monitorear y distribuir la carga entre las bases de datos remota y local. Por lo tanto, la base de datos remota (compartida) es usada solo en el caso de sobreflujo. Si a un tiempo dado, el número de llamadas almacenadas por el CP excede un cierto umbral, el CP direcciona las llamadas hacia la base de datos remota (compartida). En otras palabras, durante periodos de alto flujo de tráfico, la base de datos local y remota son usadas para el

procesamiento de una llamada que requiere servicio de red inteligente.

La figura 19, nos muestra un diagrama de tiempo del funcionamiento del mecanismo de control a dos niveles propuesto. Donde podemos ver como este mecanismo es implementado en el SSP y SCP. En este mecanismo, el SCP no requiere de hacer uso de mensajes de control específicos hacia los SSPs. El mecanismo de control usado en el primer nivel es propuesto para limitar el número de llamadas que requieren ser servidas por el SCP. Este mecanismo es necesario a fin de garantizar una buena operación del sistema y un cierto nivel en la calidad de servicio a los abonados. Nosotros proponemos el uso de un mecanismo de control que hace uso de una ventana dinámica. El propósito de este mecanismo es primero, el garantizar que una llamada sea servida al menos en t_{max} segundos. Ésto se logra limitando el número de llamadas que entran al sistema. Se adopta el principio de que es mejor informarle al abonado que el sistema no está disponible para proveer el servicio, que tener que abortar la llamada cuando ésta ya fue aceptada y se encuentra en el proceso de servicio.

El mecanismo de la ventana hace uso de un tamaño de ventana dinámico similar al usado en TCP (Tanenbaum, 1997). La ventana es inicializada a 1 y es incrementada por uno cada que una llamada es recibida como respuesta en el SSP, hasta un valor máximo de $WinMax$. Una llamada es considerada como exitosa si ésta es servida en menos de un intervalo de tiempo de t_s segundos (típicamente $t_s=2$ seg.). El tiempo es medido desde el momento que el SSP manda la llamada hacia el SCP hasta que el SSP recibe la respuesta de la llamada por el SCP. De este modo por cada llamada enviada hacia el SCP un temporizador es activado en el SSP. El mecanismo de tiempo de respuesta es implementado

similarmente al usado por TCP. Si el temporizador activado en el momento que una llamada fue enviada hacia el SCP a través de la red de señalización número 7, expira, el tamaño de la ventana (*WIN*) es decrementado, tal y como en el mecanismo de control de Ventana explicado anteriormente.

El segundo nivel es implementado en el SCP donde un control de sobreflujo es activado y desactivado comparando el número de llamadas en el sistema con dos valores de umbrales impuestos. Cuando el número de llamadas en el sistema excede al valor del umbral superior, el control de sobre flujo es activado y de este modo el CP direcciona las llamadas hacia la base de datos remota compartida (en la figura este momento se representa por un cuadro oscuro). En otras palabras, durante periodos de muy alto flujo de tráfico, la base de datos local y remota (compartida) son usadas para el procesamiento de una llamada inteligente. El CP para de enviar las llamadas hacia la base de datos remota (compartida) cuando el número de llamadas en el sistema cae por debajo del umbral inferior impuesto.

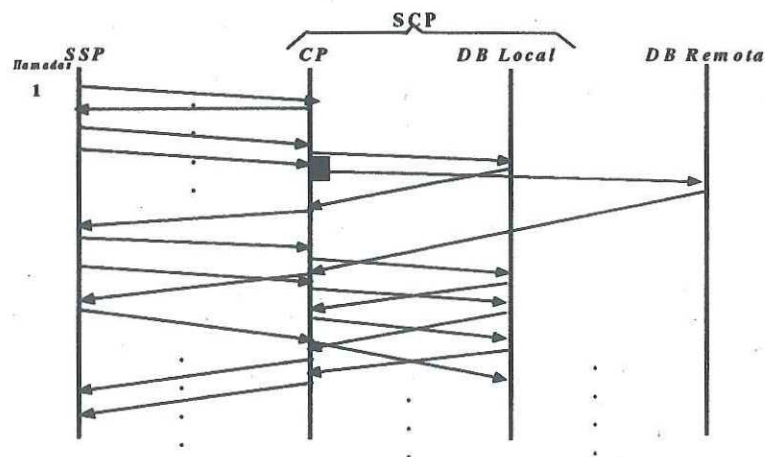


Figura 19 Diagrama de tiempo del Mecanismo de Control a dos niveles.

IV. MODELADO Y SIMULACIÓN DE MECANISMOS DE CONTROL EN LA RED INTELIGENTE

IV.1 INTRODUCCIÓN

El SCP es un elemento muy importante en la arquitectura de RI. Éste funciona como un servidor lógico en la red inteligente conteniendo la mayoría de la información y la lógica de servicio. En un futuro muy cercano, la introducción de nuevos y mas avanzados servicios de RI implicarán no solo un incremento en el tráfico de señalización, sino también un incremento en la carga de tráfico impuesta al SCP. Por lo tanto, esto introducirá problemas de cuellos de botella en los nodos centralizados (SCP) lo cual puede resultar en largos tiempos de respuesta en el servicio de una llamada inteligente y reducción en la capacidad de manejo de llamadas en la red, reduciendo con esto el caudal eficaz.

En este capítulo explicaremos los modelos de colas desarrollados para representar la arquitectura de RI y encontraremos por medio de herramientas de simulación los mejores parámetros a utilizarse en los esquemas para controlar la carga impuesta al SCP, llamados: Control de Bandas de Llamadas (Call Gapping) y Control de Ventana (Window), como también los parámetros utilizados para llevar a cabo el mecanismo de control de congestión a dos niveles para garantizar una mejor calidad de servicios (QoS, Quality of Service) de RI. En el capítulo V se evaluará la configuración de Red Inteligente bajo estudio con cada uno de los mecanismos de control propuestos haciendo uso de los parámetros obtenidos en esta sección.

V.2 DESCRIPCIÓN DEL MODELO BAJO ESTUDIO

IV.2.1 DESCRIPCIÓN GENERAL

La configuración para representar la arquitectura de la red inteligente a nivel nodo se muestra en la figura 20. En donde podemos ver que está compuesta por tres entidades físicas SSP, SCP , y RDB.

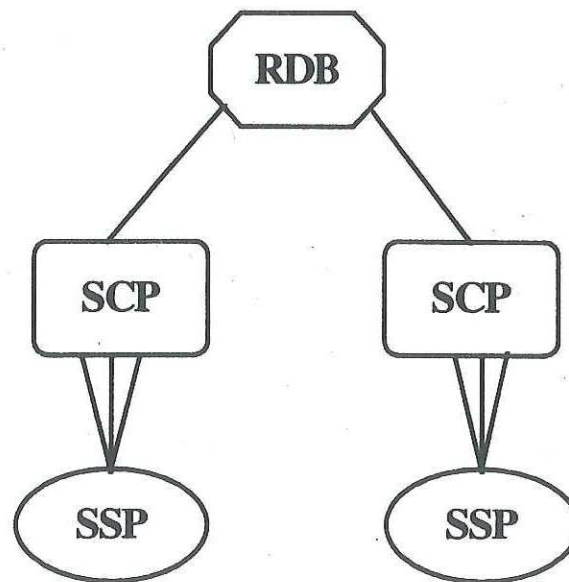


Figura 20 *Arquitectura de RI bajo estudio a nivel nodo.*

Para simplificar la explicación de nuestra configuración nos enfocaremos tan solo al uso de un SSP conectado a un SCP y a la base de datos remota. El modelo de colas utilizado para representar la configuración de la Red Inteligente es el mostrado en la figura 21. Este modelo fue creado usando el simulador OPNET¹ (OPTIMIZED NETWORK ENGINEERING TOOLS) (OPNET, 1995).

¹ Ver Apendice A, Descripción general de OPNET.

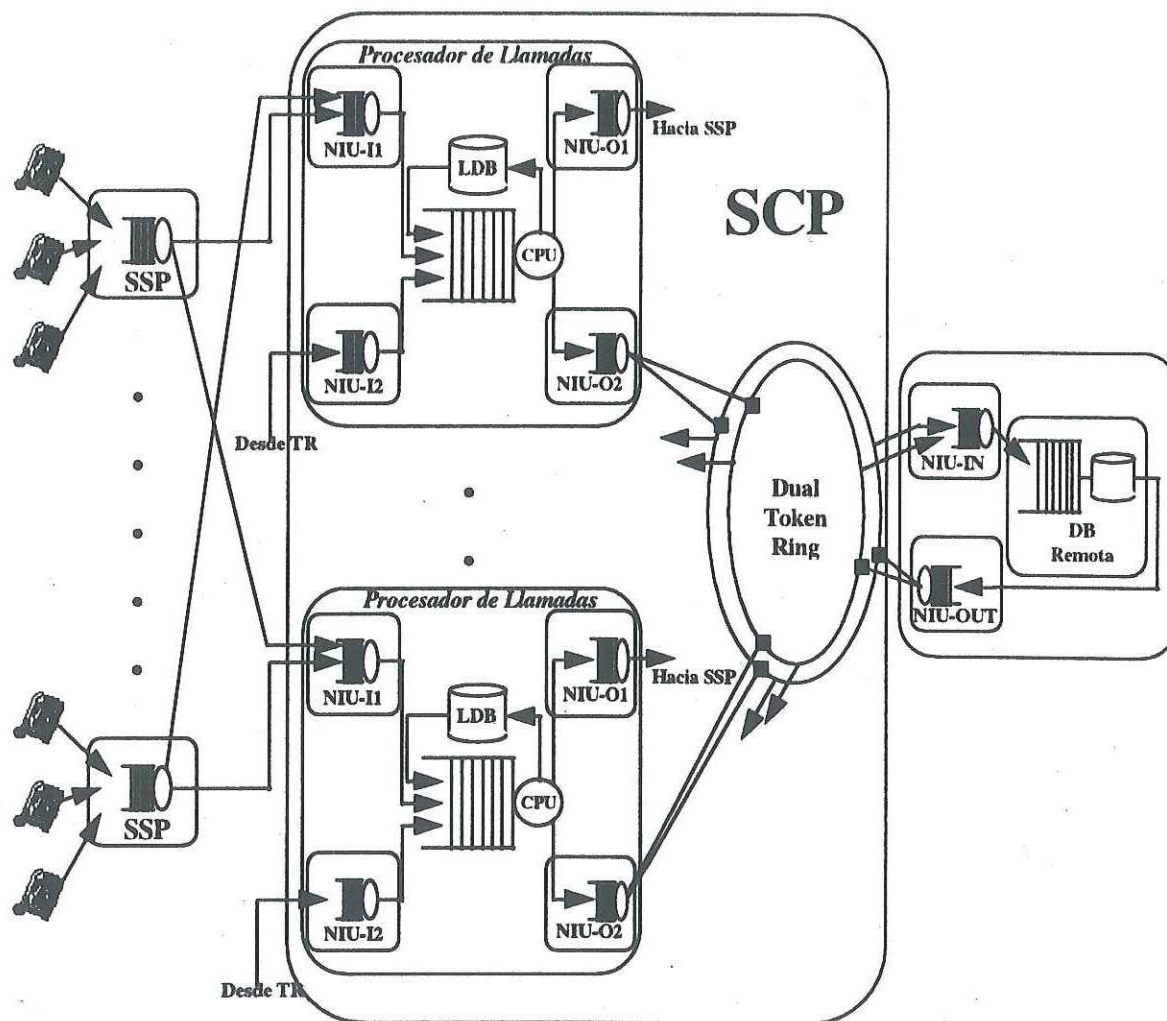


Figura 21 Modelos de colas de la Red Inteligente.

SCP: El modelo de colas utilizado para representar la configuración del SCP se muestra en la figura 21. En donde el elemento principal en el nodo SCP es una red dual Token Ring de alta velocidad. Esta topología dual de red de área local mejora la interconectividad y disponibilidad de los elementos, por ejemplo: CPs y la RDB.

Tres estaciones idénticas de CP se encuentran unidas por medio de la Token Ring dual. El tráfico emitido por los SSPs hacia el SCP es cargado equitativamente a cada CP por

medio del uso de la técnica de Round-robin. los procesadores de llamadas son los responsables de procesar los servicios inteligentes. Por lo tanto, el elemento fundamental en el SCP es el procesador de llamada. Un procesador de llamada está modelado por medio de seis subentidades llamadas NIU-I1, NIU-O1, NIU-I2, NIU-O2, CPU, LDB como se muestra en la figura 21.

NIU-I1 y NIU-O1 están representadas por un modelo de cola con capacidad de almacenamiento infinito y con un número de servidores infinitos, representando con esto un centro de retardo, el cual nos sirve para simular el tiempo que tarda una petición de servicio inteligente emitida hacia el SCP por el SSP a través de la red de señalización número 7 y viceversa. El retardo que nos proporciona la red de señalización es al menos de 100ms tomando la suposición hecha en (Robrock, 1991). Simulando en nuestro modelo que se requieren 50ms para transmitir la petición del SSP-SCP y otros 50ms para transmitir la respuesta del SCP-SSP. En nuestro modelo el retardo es simulado por medio de una distribución exponencial con media 50ms.

NIU-I2 y NIU-O2 representan las interfaces de entrada y salida a la red de área local Token Ring, respectivamente. En donde hemos hecho uso del modelo de Token Ring proporcionado por OPNET, haciendo unas modificaciones para convertir la topología sencilla de Token Ring a una red dual. Para mas detalle de Token Ring en OPNET ver (OPNET, 1995).

CPU: La unidad de procesamiento central esta modelada por una cola de tipo FIFO con buffer finito y servidor único. Cada CPU en los tres CPs que conforman la arquitectura tiene un tiempo de procesamiento de 3.33ms simulado por medio de una distribución normal

truncada a 3.75ms a la izquierda y 3.0ms a la derecha, esto quiere decir que cada CPU es capaz de procesar de 266 a 333 llamadas por segundo con una media de 300 llamadas por segundo, esto nos indica que en total el SCP es capaz de servir de 800 a 1000 llamadas por segundo con una media de 900 llamadas por segundo. Siempre y cuando las peticiones de servicio no requieran del acceso a la base de datos local ni remota.

LDB: La base de datos local esta modelada por medio de una cola FIFO con buffer finito y servidor único. En donde la longitud del buffer está compartida con el buffer de CPU en cada CP. Es decir, que la suma del buffer del CPU más el buffer de LDB nos da la capacidad total de almacenamiento del procesador de llamadas. El tiempo de servicio de la base de datos local está simulado por medio de una distribución exponencial truncada con una media de 8ms, 6ms a la derecha y 10ms a la izquierda.

2) SSP: El modelo que conforma al SSP es muy sencillo, está formado por una cola FIFO con buffer finito y servidor único, con un tiempo de servicio exponencial de 1.25ms, es decir tiene la capacidad de servir a 800 llamadas por segundo. El SSP hace uso de la técnica de Round-robin para cargar de igual manera a cada CP en el SCP. El SSP está conectado a los CPs a través de los enlaces CCS7 (Jabbari, 1991), (Chung y Sung, 1993) simulados por NIU-I1 y NIU-O1.

3)RDB: La base de datos remota esta compuesta por tres subentidades llamadas NIU-IN, NIU-OUT y Base de Datos Remota:

NIU-IN y NIU-OUT representan exactamente los mismo que NIU-I2 y NIU-O2, respectivamente. Éstas son utilizadas como interfaces a la dual Token Ring.

BD Remota: Esta modelado por una cola FIFO con buffer finito y servidor único el cual simula el tiempo de servicio que conlleva la consulta de la información. El tiempo de servicio es simulado por medio de una distribución exponencial truncada con una media de 8ms con 6ms a la derecha y 10ms a la izquierda.

FLUJO DE LLAMADAS EN LA RED.

El SSP detecta una llamada inteligente la cual es enviada al SCP a través de la red SS7 para ser servida. Las llamadas inteligentes son almacenadas temporalmente en la interfaz que simula el tiempo de retardo proporcionada por la red de señalización colocadas en cada uno de los CPs (NIU-II). Después, la llamada inteligente es almacenada en el buffer de CPU en donde espera antes de ser servida. El CPU procesa la llamada inteligente verificando si la información requerida por la llamada para llevar a cabo el servicio inteligente se encuentra almacenada en memoria principal del procesador de llamadas o si se requiere de información almacenada en la base de datos local. Si se requiere el acceso a la base de datos local, la petición de servicio inteligente es enviada a la base de datos local en donde es almacenada temporalmente antes de ser servida, una vez servida la petición por la base de datos local esta es regresada de nuevo al CPU donde se vuelve almacenar la respuesta mandada por la base de datos local en el buffer del CPU, de nuevo el CPU procesa la respuesta y verifica si la información es suficiente para llevar a cabo el servicio satisfactoriamente por el SSP, si la llamada inteligente aun no es completada por falta de información el procesador verifica si se requiere de un nuevo acceso a la base de datos local para tomar la información restante, si esto es así la llamada inteligente es enviada de nuevo a consultar la base de datos local, una vez consultada la base de datos local, de nuevo la

petición de servicio es almacenada en el buffer del CPU en espera de ser nuevamente procesada, una vez que la petición es procesada, el CPU verifica si la información es la necesaria para llevar a cabo el servicio inteligente. Si la información es la indicada se envía la llamada inteligente con la información necesaria hacia el SSP por medio de la interfaz que simula el retardo de la red de señalización (NIU-O1) (un ejemplo de este tipo de flujo de información es requerido cuando se lleva a cabo el servicio llamado Credit Card Calling). De otro modo, si la información requerida para llevar a cabo el servicio inteligente se encuentra en memoria principal, el procesador de llamada envía la respuesta a la llamada inteligente directamente después de procesarla hacia el SSP, a través de la interface NIU-O1 (Un ejemplo de este tipo de flujo de información es requerido cuando se lleva a cabo el servicio llamado Freephone).

Por otra parte, la base de datos remota (compartida) es usada cuando los procesadores de llamadas direccionan llamadas inteligentes hacia ésta. (La base de datos remota (compartida) es usada en el caso de sobre flujo o cuando la base de datos local de un procesador de llamadas falle). Una vez que el procesador de llamadas direcciona una llamada inteligente hacia la base de datos remota, la llamada es almacenada temporalmente en la interface de la Dual Token Ring (NIU-O2) en donde en forma aleatoria se escoge el anillo por el cual la petición se va enviar hacia la base de datos remota. Una vez que la llamada inteligente llega a la base de datos remota esta es almacenada temporalmente por la interfaz de entrada de la Dual Token Ring (NIU-IN) colocada en la base de datos remota, después la petición de servicio es enviada al buffer donde la petición espera ha ser servida, el tiempo de procesamiento simula el tiempo de consulta de información en la base de datos

remota. Una vez procesada la llamada esta es enviada hacia el CP que direcciono la llamada inteligente a través de la interface de salida de la Dual Token Ring (NIU-OUT). Al llegar la respuesta de la consulta de la base de datos remota a través de la interface NIU-I2 es enviada al buffer del CPU donde espera de nuevo a ser atendida y procesada para completar el servicio. Una vez completado el servicio, el procesador de llamada envía la respuesta hacia el SSP a través de la interface NIU-O1 para que el SSP lleve a cabo el servicio inteligente. Todo este flujo de información para llevar a cabo un servicio inteligente debe llevarse a cabo en menos de 2 segundos.

La tabla III, muestra los parámetros generales utilizados para simular el modelo de colas que forma nuestra configuración de la Red Inteligente bajo estudio.

Tabla III Parámetros Generales utilizados para modelar a la Red Inteligente.

<i>Descripción de Parámetros</i>	<i>Valores usados</i>
Tiempo de Servicio de Base de Datos Local Tiempo de Servicio de Base de Datos Remota (exponencial truncada)	8ms (media) 6ms (derecha) 10ms (izquierda)
Tiempo de Servicio del SSP (exponencial)	1.25ms (media)
Tiempo de Servicio del CPU en cada CP (Normal truncada)	3.33ms(media) 3.0ms (derecha) 3.750ms (izquierda)
Interface de la Red de Señalización número 7 Tiempo de Servicio (NIU-I1 & NIU-O1) (exponencial)	50ms (media)
Unidades de Interface de Red Token Ring Tiempo de Servicio NIU-I2 & NIU-O2, NIU-IN & NIU-OUT (exponencial)	25μs (media)
Velocidad de Token Ring	16 Mbps
Número de Procesadores de Llamadas por SCP	3

IV.2.2 CON CONTROL DE BANDA DE LLAMADAS (CALL GAPPING)

La configuración a bloques para representar la arquitectura de RI en el caso de control Call Gapping se muestra en la figura 22. Las características básicas que conforman los modelos de colas del SSP, SCP y RDB son esencialmente las mismas explicadas en el punto anterior, con la excepción de que ahora se impone a la configuración dos entidades las cuales son las encargadas de llevar a cabo el control de Call Gapping.

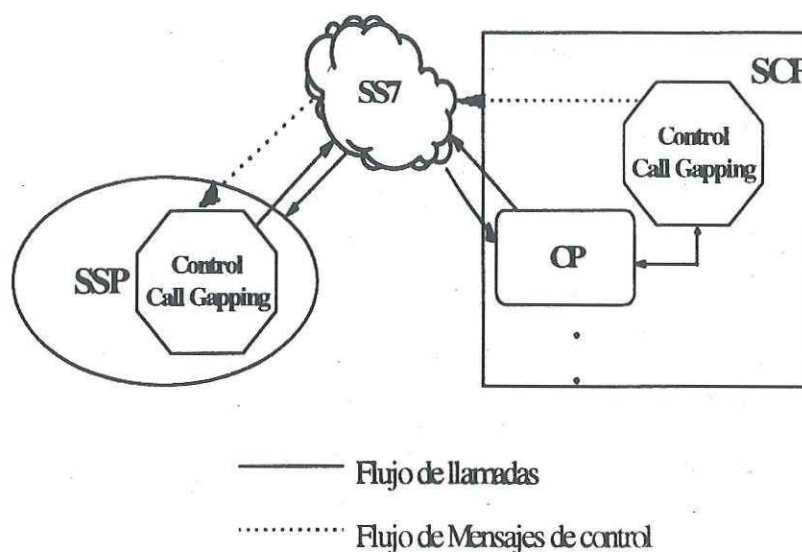


Figura 22 Configuración a bloques de RI con control de Call Gapping.

El funcionamiento básico de Call Gapping se explico en III.5.1. Sin embargo, en esta sección se explicará en forma mas general como el control afecta a cada una de las entidades que conforman la configuración de red inteligente bajo estudio.

1) Control de Call Gapping en el SCP. Cada uno de los CPs que componen al SCP está sujeto a muestreos periódicos por el mecanismo de control Call Gapping. Como se explicó anteriormente al inicio de cada intervalo de muestreo el número de llamadas que se encuentran dentro del buffer del CP en espera de ser atendidas es comparado con el valor de dos umbrales, donde si la cantidad de llamadas monitoreadas sobrepasa el umbral superior, el control es activado. En el momento que el control es activado, se simula un tiempo de procesamiento el cual es utilizado para el cálculo de estadísticas en donde se mide el tiempo medio entre salidas durante el período de muestreo, estimando con este valor el tiempo medio entre entradas y utilizando la ecuación básica (1) para calcular el valor de la banda g.

Después de obtener el valor de la banda g se crea un mensaje de control especial en donde se le informa al SSP del valor de g el cual tendrá que ajustar el flujo de salidas en el SSP. El mensaje de control es procesado y mandado hacia el SSP a través de la interfaz NIU-O1 la cual simula el tiempo de retardo proporcionado por la red de señalización número 7. Una vez mandado el mensaje de control el SSP lo detecta y extrae el valor de la banda g el cual gobernará el tiempo entre salidas en el SSP hacia el SCP. Por otra parte, si la cantidad de llamadas en espera de ser atendidas en el CP es menor o igual al umbral inferior y si el mecanismo de control no había sido activado anteriormente, el sistema permanece estable, sin embargo, si el control ya había sido activado, el CP calcula de nuevo un valor de banda g de la misma forma al explicado anteriormente para tratar de normalizar el tiempo entre salidas del SSP, creando de nuevo un mensaje de control y enviado al SSP a través de la red de señalización.

Para seleccionar los valores óptimos de los umbrales y el período de muestreo, se consideraron dos restricciones:

- a) Minimizar la probabilidad de bloqueo en el SCP.
- b) Minimizar la probabilidad de bloqueo en el SSP.

En primer lugar antes de conocer el valor óptimo de los umbrales, es necesario conocer la capacidad óptima de los buffers del sistema. Para llegar a esto se consideró que toda llamada inteligente debe de ser servida en un tiempo no mayor de 2 segundos. Por lo tanto, suponiendo un valor fijo del buffer en el SSP de 200 llamadas y considerando que se utilizará un escenario en donde el 25% de las llamadas no requieren el acceso a la base de datos local, suponiendo que la información necesaria se encuentra en memoria principal del

procesador de llamada. El 50% requiere de un acceso a la base de datos y el resto 25% de dos accesos, además suponiendo que la arquitectura no hace uso de la base de datos remota y considerando a la red en momentos de alto tráfico por medio de la ecuación (2) podemos calcular el valor óptimo de la capacidad del buffer en el SCP, considerando que el SCP es el que ocasiona el cuello de botella en el sistema.

$$t_{max} = B_{ssp} * t_{ssp} + R_{ss7} + B_{cp} [P(0)*t_{cpu} + P(1)*(2*t_{cpu}+t_{ldb}) + P(2)*(3*t_{cpu}+2*t_{ldb})] \quad (2)$$

En donde t_{max} es el tiempo requerido para cumplir la petición de servicio inteligente, B_{ssp} el tamaño del buffer en el SSP, t_{ssp} el tiempo de servicio en el SSP, R_{ss7} el retardo proporcionado por la red de señalización número 7, B_{cp} el tamaño del buffer en cada CP, $P(0,1,2)$ como la probabilidad de que la petición de servicio inteligente requiera 0, 1, 2 accesos a la base de datos, t_{cpu} como el tiempo de servicio en el CP y t_{ldb} el tiempo de servicio de la base de datos. Así, suponiendo los siguientes valores:

$t_{max} = 2s$, $B_{ssp} = 200$ llamadas, $t_{ssp} = 1.25ms$, $R_{ss7} = 100ms$, $P(0,1,2) = 25\%$, 50% y 25% respectivamente, $t_{cpu} = 3.33ms$, y $t_{ldb} = 10ms$. podemos calcular a B_{cp} el cual es considerado como el punto que nos proporciona el cuello de botella en el sistema. Entonces tenemos que:

$$B_{cp} = 99 \text{ llamadas.}$$

Por lo tanto, la capacidad el buffer en cada CP para que el retardo no exceda de 2 segundos es de 99 llamadas, utilizaremos un buffer de 100 llamadas para redondear el valor. Así, conociendo el valor óptimo para el buffer en el CP podemos continuar la búsqueda de los valores de umbrales óptimos y del tiempo de intervalo de muestreo.

Por medio de simulación y utilizando los parámetros generales vistos en la tabla III (IV.2.1), utilizando varios valores supuestos de intervalos de muestreo (0.15, 0.25, 0.5, 1 y 2 segundos) y de la misma forma suponiendo varios valores de umbrales (superior/inferior : 98/90, 90/50, 98/50, 80/50 y 90/70), encontraremos los mejores valores posibles.

En la figura 23, podemos ver los resultados obtenidos de las simulaciones graficando la probabilidad de bloqueo en el SSP con respecto al intervalo de muestreo y al valor de umbral. De la misma forma en la figura 24, se presenta la probabilidad de bloqueo en el SSP con respecto al intervalo de muestreo y al valor de umbral supuesto.

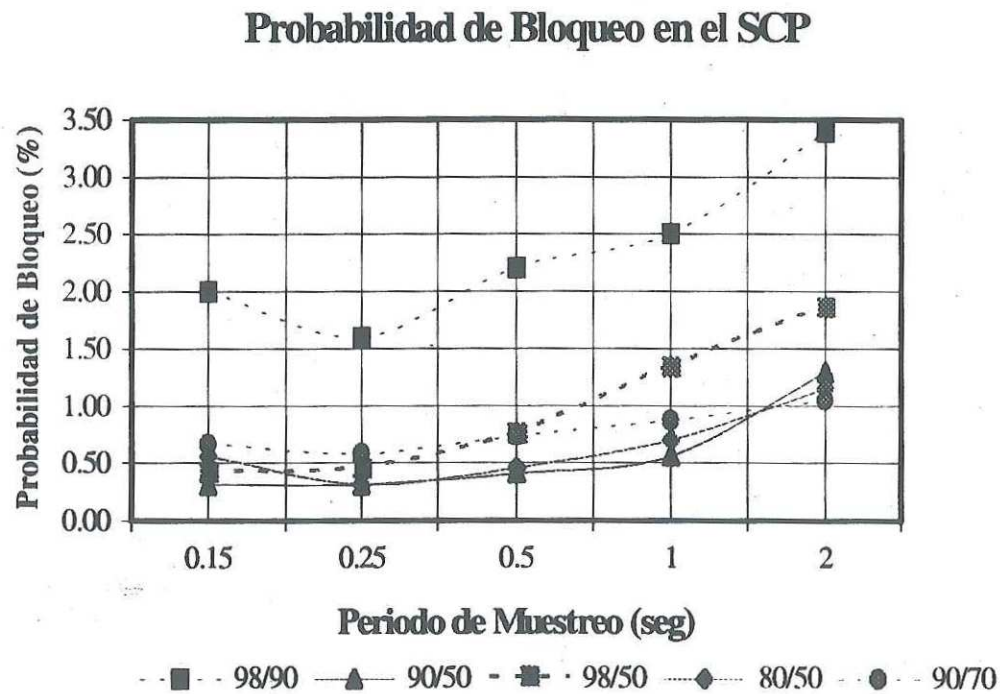


Figura 23 Probabilidad de Bloqueo en el SCP.

Probabilidad de Bloqueo en el SSP

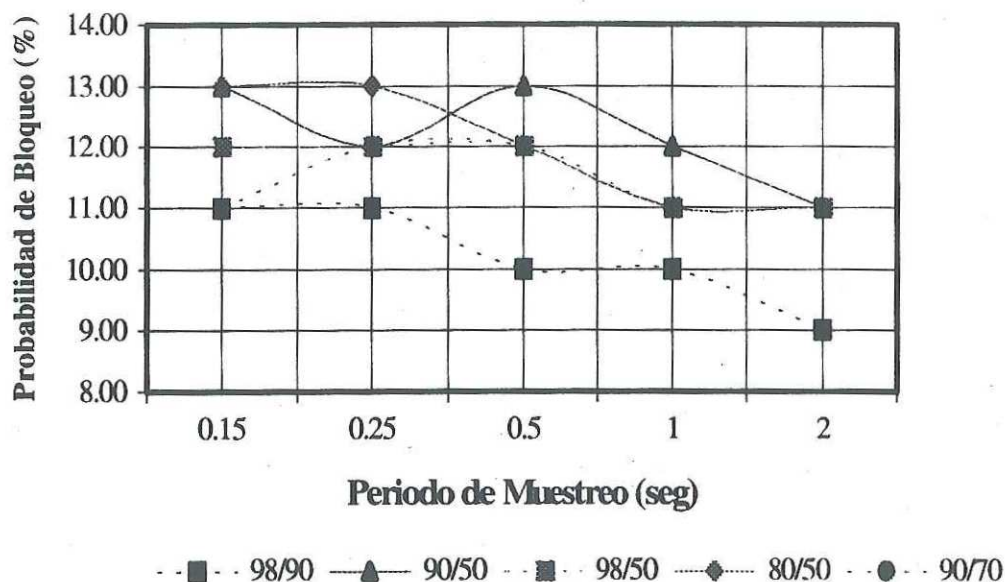


Figura 24 Probabilidad de Bloqueo en el SSP.

Nos referimos a llamadas bloqueadas en el SCP como aquellas llamadas que fueron previamente aceptadas por el SSP y enviadas como peticiones de servicio hacia el SCP en espera de ser servidas, y estas fueron bloqueadas por el desbordamiento del buffer del CP en periodos de alto tráfico. De la misma forma nos referimos a llamadas bloqueadas en el SSP como aquellas llamadas que desde el inicio no fueron aceptadas por el SSP y que rápidamente el abonado pudo ser avisado de la falta de recursos disponibles en la red. Considerando la suposición de que es mejor que el abonado sea avisado desde el inicio de la falta de recursos en la red a que este sea atendido sin ser servido. De este modo los valores de los umbrales y el valor del período de muestreo serán considerados los mejores cuando

la probabilidad de bloqueo en el SCP sea la mínima y buscando que la probabilidad de bloqueo en el SSP se compense relativamente.

Por lo tanto de la figura 23, podemos observar que la probabilidad de bloqueo en el SCP es mínima cuando se usa un tiempo de muestreo de 0.25 segundos. De la misma figura vemos que si el período de muestreo es menor es decir mas constante, el CPU en el CP está mas tiempo ocupado calculando las estadísticas necesarias para llevar a cabo el control de Call Gapping y con esto permitiendo que el número de llamadas que residen en el buffer en espera de ser atendidas incremente y la probabilidad de bloqueo en el SCP aumente respectivamente. De la misma figura podemos observar que los valores de umbrales considerados los mejores son el de 90/50 y 80/50 en este punto (0.25seg.). Por lo tanto, ya tenemos el mejor valor para el período de muestreo con respecto a la probabilidad de bloqueo en el SCP, pero necesitamos escoger los mejores valores de umbrales. Para esto, observamos la figura 24, la probabilidad de bloqueo en el SSP en donde al punto del valor del período de muestreo seleccionado anteriormente y observando el comportamiento de los valores de los umbrales seleccionados, podemos ver que los valores 90/50 nos permite una menor probabilidad de bloqueo en el SSP a diferencia de 80/50, de este modo podemos seleccionar los valores 90/50 como los mejores, considerando que estos valores compensan la probabilidad de bloqueo en el SSP con respecto a la probabilidad de bloqueo en el SCP.

2) Control de Call Gapping en el SSP. El SSP está formado de la misma manera como se explicó anteriormente por un modelo de cola con buffer finito y servidor único. Haciendo uso de una técnica de round-robin para cargar de igual manera a cada CP que forman un SCP. Sin embargo, ahora se ve afectado por un mecanismo de control el cual al momento

de que el SSP detecta un mensaje de control enviado por el SCP en momentos de alto tráfico, este modifica el tiempo entre salidas de peticiones enviadas hacia el SCP. Es decir, el mensaje de control nos proporciona un valor de banda g que gobierna el tiempo entre salidas, como lo muestra la figura 25. El tiempo de salida se normaliza en el momento que la carga en el SCP baja de un cierto nivel de umbral o sigue en continuo cambio si la carga en el SCP sobre pasa un umbral superior.

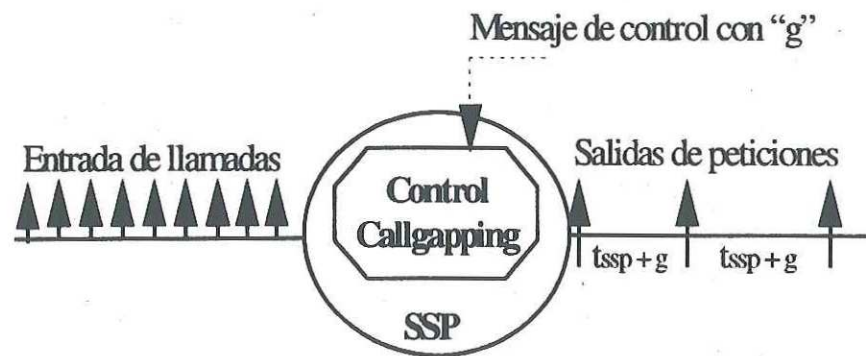


Figura 25 Reacción del mecanismo de control Call Gapping en el SSP

De este modo, podemos presentar los parámetros que serán utilizados para las simulaciones finales de nuestro estudio en el caso del mecanismo de control de bandas de llamadas Call Gapping, los cuales se presentan en la tabla IV.

Tabla IV Parámetros para el Mecanismo de Control Call Gapping.

<i>Descripción de parámetros</i>	<i>valores usados</i>
Umbral Superior (Ts)	90
Umbral Inferior (Ti)	50
Periodo de Muestreo	0.25 seg.
Nueva g	$T_{out} = g + T_{in}$

IV.2.3 CON CONTROL DE LA VENTANA (WINDOW)

La configuración a bloques para representar la configuración de red inteligente con control de la ventana se muestra en la figura 26. Los modelos de colas de cada uno de los componentes que conforman la arquitectura son modelados básicamente de la misma manera como se explico en IV.2.1 pero con la diferencia de que ahora el sistema hace uso de un mecanismo que monitorea el comportamiento del SSP.

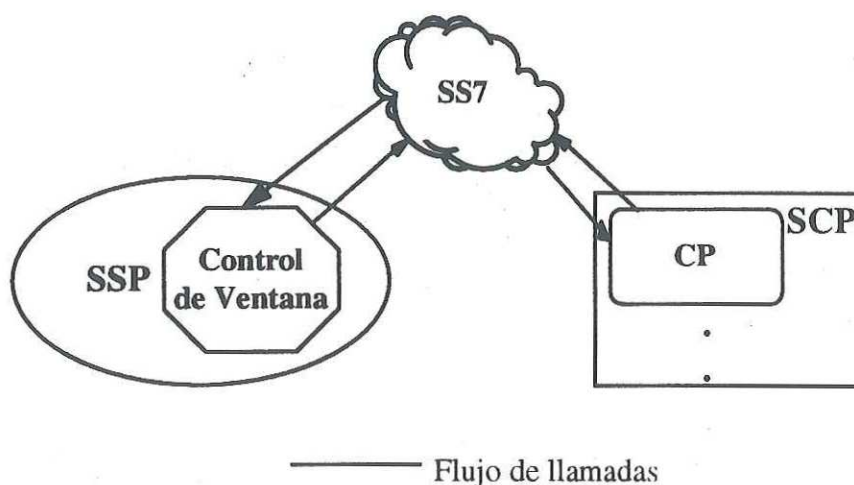


Figura 26 Configuración a bloques de RI con Control de la Ventana.

El comportamiento básico del mecanismo de control de la Ventana lo podemos ver en III.5.2. Sin embargo en esta sección nos enfocaremos mas en detalle el papel que juega el mecanismo de control de la Ventana en cada uno de los elementos que conforman la configuración.

1) Control de Ventana en el SSP. Principalmente en la arquitectura de Red Inteligente con control de la Ventana el SSP debe de estar informado con anterioridad del número máximo de llamadas que puede enviar y que deben de ser cumplidas dentro de los límites de tiempo

establecido (2s). En donde el número máximo representa el valor de la ventana máxima (WinMax) que el SSP puede alcanzar. En momentos de alto flujo la ventana máxima se convierte en la capacidad máxima del buffer del destino en este caso el SCP. De esta manera usando las mismas suposiciones hechas en Call Gapping para encontrar el tamaño del buffer óptimo del SCP se hace uso de la ecuación (2) en donde obtenemos que el valor del buffer en el SCP es de 99 llamadas en momento de alto flujo en la red. Así, WinMax toma el valor de 99 llamadas, el cual será redondeado a 100 llamadas. De este modo, el SSP no podrá tener mas de 100 llamadas en espera de ser servidas en un momento dado, con esto se protege que las llamadas no alcancen un tiempo de respuesta mayor a 2 segundos y se protege automáticamente al destino de posible congestión.

El SSP hace uso de temporizadores para llevar a cabo el mecanismo de control, activando un temporizador cada vez que una petición de servicio inteligente es enviado hacia el SCP a través de la red SS7. El temporizador es inicializado a una longitud de 2 segundos el mismo valor requerido para que una respuesta a la petición sea exitosa. Si el temporizador se activa nos indica que el destino SCP ha caído en un posible estado de congestión o que la llamada se ha perdido. De otra forma si la llamada es atendida positivamente antes que el temporizador se active, éste es desactivado automáticamente en el momento en que la respuesta a la petición de servicio inteligente enviada por el SCP llega al SSP.

El temporizador y la respuesta a la petición de servicio enviada por el SCP nos ayudan a establecer dos políticas usadas por este mecanismo de control, la primera es la política de incremento del tamaño de la ventana y la segunda la política de decremento del

tamaño de la ventana. Para encontrar la mejor manera para llevar a cabo estas políticas se hizo uso de simulación en donde para:

a) Política de Incremento

- 1) La ventana incrementa en uno cada vez que llegan tres respuestas enviadas por el SCP a peticiones de servicio inteligente al SSP.
- 2) La ventana incrementa en uno cada vez que llegan dos respuestas enviadas por el SCP a peticiones de servicio inteligente al SSP.
- 3) La ventana incrementa en uno cada vez que llega una respuesta enviada por el SCP a peticiones de servicios inteligentes al SSP.

El resultado de estas simulaciones lo podemos observar en la tabla V, en donde N es el número de respuestas a llamadas inteligentes esperadas para incrementar la ventana en uno. La probabilidad de bloqueo en el SSP nos representa el número de llamadas que fueron bloqueadas por el SSP debido al desbordamiento del buffer. Vemos que entre mayor es N la probabilidad de bloqueo aumenta, como lo esperábamos. Debido a que el tamaño de la ventana aumenta muy lentamente permitiendo que las llamadas se almacenen muy rápido en el buffer del SSP y en alto flujo de tráfico la probabilidad de bloqueo aumente. Por otra parte si $N=1$, en donde la ventana se incrementa a uno cada vez que se recibe una respuesta de petición de servicio inteligente, el valor de la ventana incrementa permitiendo un mayor tráfico, y llegando al máximo valor del tamaño de la ventana rápidamente. Entonces podemos concluir que el mejor valor de incremento de la ventana es $N=1$. En la figura 27, se muestra el comportamiento del mecanismo en el SSP, en donde el tamaño de la ventana se incrementa en uno por cada respuesta a una petición de servicio hecha al SCP, hasta

alcanzar un valor máximo (WinMax).

Tabla V Resultados obtenidos con la utilización de diferentes políticas de incremento.

Política Valor de N	Probabilidad de Bloqueo en el SSP
1	0.10
2	0.12
3	0.13

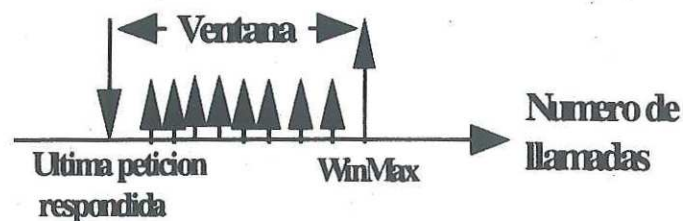


Figura 27 Comportamiento del mecanismo de Control de la Ventana en el SSP.

b) Política de Decremento

- 1) El valor de la ventana decrementa hasta el valor mínimo igual a 1, cuando ocurre la activación de un temporizador en el SSP (Decremento Repentino).
- 2) El valor de la ventana decrementa en uno, cuando ocurre la activación de un temporizador en el SSP (Decremento Gradual).
- 3) El valor de la ventana decrementa a la mitad y redondeado a un entero al ocurrir la activación de un temporizador en el SSP (Decremento Binario).

Para investigar la mejor política de decremento se utilizaron los parámetros generales dados en la tabla III, y suponiendo un valor de ventana máximo de 120 llamadas

para provocar el desbordamiento del buffer en el SCP en periodos de alto flujo de tráfico y con ésto la activación del temporizador en el SSP. Permitiendo medir la probabilidad de bloqueo en el SCP y en el SSP como parámetros claves para la selección de la política óptima. En la tabla VI presentamos los resultados de cada uno de este conjunto de simulaciones previas.

Tabla VI Resultados obtenidos con la utilización de diferentes políticas de decremento.

Política	Probabilidad de Bloqueo en el SCP	Probabilidad de Bloqueo en el SSP
Decremento Repentino	0.0019	0.12
Decremento Gradual	0.0020	0.11
Decremento Binario	0.0022	0.13

De estos resultados, podemos concluir que decrementar la ventana al mínimo nos proporciona una menor probabilidad de bloqueo en el SCP pero no la menor probabilidad de bloqueo en el SSP. En nuestro estudio, tomaremos la política de decremento gradual puesto que nos proporciona una buena probabilidad de bloqueo en el SSP. Esto debido a que utilizamos el valor del tamaño máximo de la ventana igual al valor del tamaño del buffer en el SCP, con esto protegiendo al SCP de posibles estados de congestión y no esperamos tener un número de temporizadores activados continuamente. Solo unos pocos debido al redondeo del valor máximo del buffer y a las variaciones que nos proporcionen las distribuciones de servicios que simulan el tiempo de servicio por cada entidad en la red. Con esto alcanzando la menor probabilidad de bloqueo en el SSP.

Por otra parte, la inicialización del valor de la ventana es colocada a uno puesto que este no es un parámetro que afecte al desempeño del sistema en el transcurso del tiempo de simulación.

2) Control de la Ventana en el SCP. El mecanismo de la ventana no afecta en lo mas mínimo al SCP. Por lo tanto, la configuración del SCP es la misma a la utilizada cuando la configuración no tiene un mecanismo de control.

En la tabla VII, podemos resumir los parámetros que serán utilizados además de los parámetros generales dados en la tabla III para las simulación final de nuestro estudio de la configuración de Red Inteligente haciendo uso del mecanismo de control de la Ventana (Window).

Tabla VII Parámetros para el mecanismo de control de la Ventana en RI.

<i>Descripción de Parámetros</i>	<i>Valores usados</i>
Tamaño máximo de Ventana (WinMax)	100
Tamaño de la Ventana Inicial (WIN)	1
Valor de Temporizador en tiempo de respuesta (Ts)	2seg.

IV.2.4 CON CONTROL A DOS NIVELES (TWO LEVELS CONTROL)

La configuración a bloques para representar la configuración de red inteligente con el mecanismo de control de congestión a dos niveles, es mostrada en la figura 28. En donde podemos observar que la implementación del mecanismo se hace en las dos entidades físicas que conforman la arquitectura estas son el SSP y el SCP.

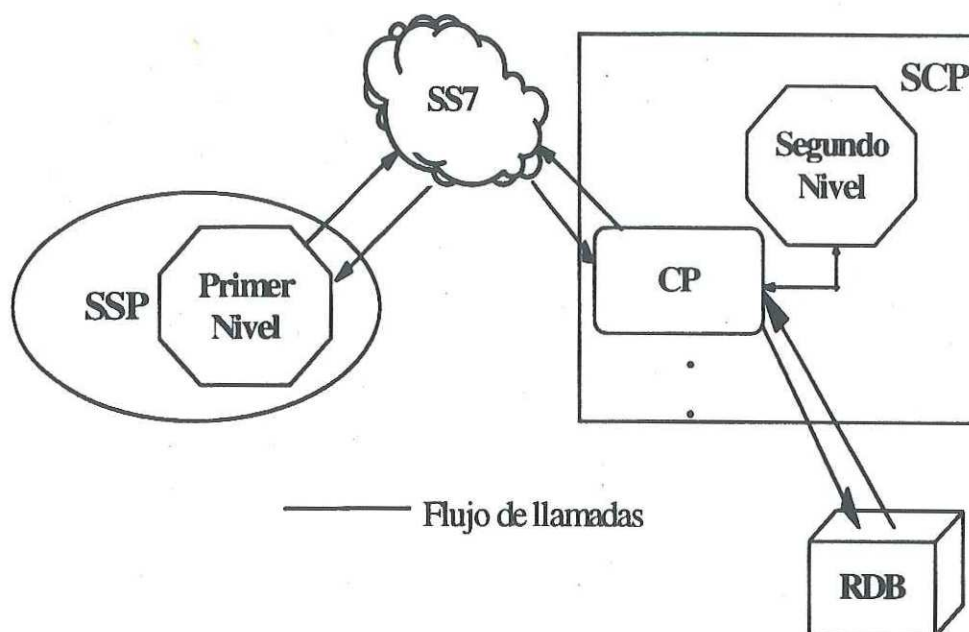


Figura 28 Configuración a bloques de RI con Control de Congestión a dos niveles.

El comportamiento básico del mecanismo de control de congestión a dos niveles fue visto en III.5.3. Sin embargo, en esta sección nos enfocaremos con mas detalle a como cada mecanismo es llevado a cabo en cada una de las entidades físicas que conforman la arquitectura.

- 1) Control de congestión a dos niveles en el SSP. El control que se lleva a cabo en el SSP es el primer nivel de este mecanismo, en donde se hace uso del mecanismo de control de la ventana visto en IV.2.3 (utilizando los mismos parámetros dados en la tabla VII).
- 2) Control de congestión a dos niveles en el SCP. El mecanismo de control en el SCP conforma el segundo nivel, en donde se hace uso del valor de dos umbrales, los cuales se detectan en forma automática al momento que llega una petición de servicio inteligente hecha por el SSP a través de la red de señalización. Este mecanismo en el segundo nivel es llamado control de sobreflujo. Al momento de que una petición llega al SCP,

automáticamente se compara el número de peticiones que se encuentran en el buffer del CP en espera de ser atendidas con dos valores de umbrales. En este mecanismo de sobre flujo en el momento que el número de llamadas inteligentes sobre pasa el umbral superior el CP empieza a direccionar las peticiones de servicio hacia la base de datos remota (compartida) con el fin de sobrellevar la cantidad de tráfico sobre impuesto al SCP, de esta manera permitiendo que el caudal en el SCP, el tiempo de respuesta a la petición de servicio inteligente y la probabilidad de bloqueo en el SSP sean optimizados. De la misma forma cuando el número de llamadas inteligentes en el buffer de cada CP es menor al valor del umbral inferior el CP para de direccionar el envío de peticiones hechas a la base de datos remota (compartida). Por lo tanto en momentos de alto tráfico, la base de datos local y la base de datos remota son usadas en forma paralela para el procesamiento de una llamada inteligente. Entonces podemos considerar que el segundo nivel del mecanismo es un conmutador el cual es encendido solo en momentos de alto tráfico direccionando llamadas inteligentes hacia la base de datos remota (compartida). Adicionalmente, este conmutador podría ser utilizado en momentos cuando la base de datos local de un CP falle.

El mecanismo de control de congestión a dos niveles hace uso del mecanismo de la Ventana en el primer nivel y un control de sobre flujo en el segundo nivel, como también a diferencia de los otros mecanismos hace uso de la base de datos remota (compartida) propuesta sólo para llevar a cabo el estudio del mecanismo de control de congestión a dos niveles. La tabla VIII, nos proporciona los parámetros utilizados por el mecanismo de control de congestión a dos niveles.

Tabla VIII *Parámetros para el mecanismo de control de congestión a dos niveles en RI.*

<i>Descripción de Parámetros</i>	<i>Valores usados</i>
Primer Nivel	Control de la Ventana (Tabla VII)
Segundo Nivel:	
Umbral superior	80
Umbral inferior	50

V. ANÁLISIS DE RESULTADOS

V.1. INTRODUCCIÓN

En este capítulo llevaremos a cabo el análisis de resultados obtenidos de simular la configuración de red inteligente propuesta, haciendo uso del simulador OPNET en el que se elaboró el modelo bajo estudio.

El objetivo principal de nuestro estudio es el de proteger al SCP de sobreflujo, lo cual decrementa el caudal eficaz en el SCP e incrementa el tiempo de respuesta en el sistema. Por lo tanto, nuestro análisis se llevará a cabo de tal manera que nuestro objetivo quede cubierto con los mejores resultados posibles. En esta sección analizaremos el modelo de tráfico utilizado para llevar a cabo las simulaciones, también identificaremos el elemento en nuestra configuración que representa el problema de cuello de botella en el sistema, por medio del estudio del comportamiento de los CPs, monitoreando el buffer, y la utilización del CPU y la LDB que conforma cada uno de los CPs. Después analizaremos el tiempo de respuesta a una llamada inteligente desde el momento en que el SSP envía la llamada hacia el SCP, hasta el momento en que la respuesta arriba al SSP. Por último estudiaremos el caudal eficaz en el sistema representado por el Goodput en el SCP.

Nuestro análisis será dividido en dos fases en donde la primera fase (FASE I) está formada por un grupo de simulaciones en donde el objetivo principal es el de establecer las bases que nos permitan comparar el desempeño del sistema sin control y utilizando los mecanismos de control Call Gapping y Ventana. Un segundo grupo de simulaciones conforma la segunda fase de nuestro análisis (FASE II) en donde el objetivo principal es

explorar el uso de una base de datos remota compartida por todos los CPs que conforman al SCP. La base de datos remota es usada como una propuesta para disminuir el tiempo de respuesta en las llamadas inteligentes y aumentar el caudal eficaz en el SCP manteniendo un Goodput considerable y disminuyendo la probabilidad de bloqueo. El uso de una base de datos remota requiere un mecanismo de control capaz de mantener al sistema fuera de posibles estados de congestión, por lo tanto, en esta segunda fase se estudiará el mecanismo de control de congestión a dos niveles en donde la base de datos remota es usada en momentos de sobreflujo en los CPs.

El análisis será enfocado a un solo CP de los tres que conforman al SCP, considerando que cada uno de estos está cargado de la misma manera y por lo tanto el desempeño de cada uno de ellos es similar. Esto para simplificar el análisis.

V.2 MODELO DE TRÁFICO

El modelo de generación de tráfico en un estudio de simulación es uno de los parámetros mas importantes. Para nuestra simulación hemos construido un generador de tráfico basándonos en fluctuaciones de tráfico real. En donde el tráfico fue obtenido gracias al monitoreo de la Red Inteligente de Stentor Resource Center, Ottawa, Canadá. En un período del 16 al 21 de febrero de 1997, comenzando en domingo y terminando en viernes. Los datos monitoreados de tráfico son representados en una gráfica, con el fin de ver y entender el comportamiento de tráfico de servicios inteligentes. La gráfica está compuesta por 120 puntos los cuales representan la cantidad de llamadas por unidad de tiempo para una hora del período de monitoreo. La figura 29, nos muestra el comportamiento del

tráfico de Servicios Inteligentes monitoreado en la red arriba mencionada.

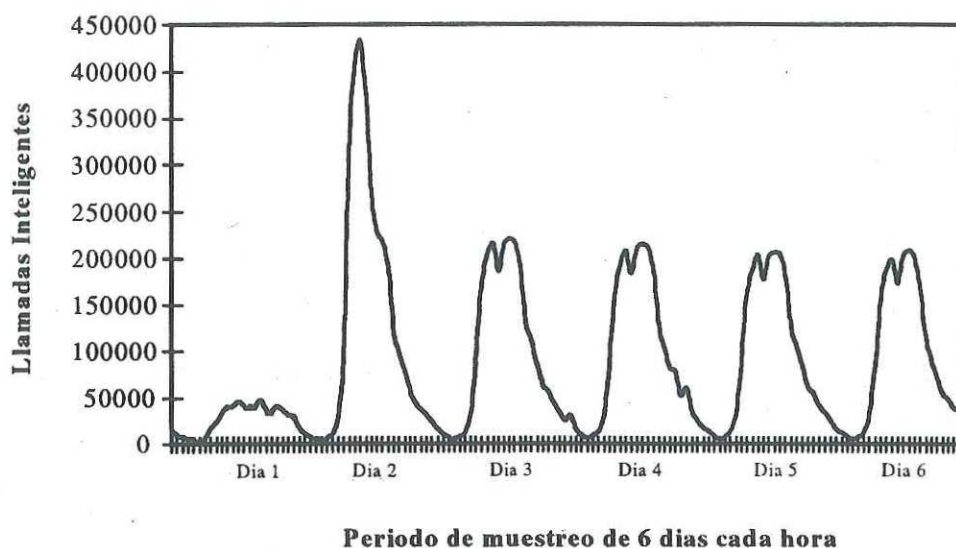


Figura 29 Tráfico monitoreado en la red inteligente en Stentor Resource Center.

Los servicios inteligentes monitoreados son servicios de llamadas 800, 888 y el servicio llamado Service Finder (Encontrar Servicio). Como podemos notar en la gráfica del tráfico real, el número de llamadas por segundo máximo alcanzado es de aproximadamente 125 llamadas en el día 2.

Nuestro modelo de tráfico fue creado de tal manera que simulemos las posibles fluctuaciones de la realidad, incrementando el tráfico hasta sobrepasar la capacidad del sistema, de este modo podemos evaluar el desempeño de la red con la implementación de los diferentes mecanismos de control de flujo bajo estudio. Entonces se llevo a cabo lo siguiente:

- 1) Se tomaron dos días de la semana para la creación del modelo de tráfico (martes y miércoles: día 3 y 4).

- 2) Se agruparon las 48 horas de los dos días en pares, formando así 24 variaciones de tráfico.
- 3) Se sacó la media de llamadas cada 2 horas del tiempo real (media de llamadas por dos horas).
- 4) Se convirtió la media de llamadas por dos horas a una media de llamadas por segundo en dos horas, obteniendo 24 valores que nos representan la media de llamadas por segundo en dos horas.
- 5) Con el fin de llevar al sistema a su capacidad máxima, se consideró nueve veces la media de llamadas por segundo en dos horas a un segundo de tiempo de simulación. En donde en nuestro modelo de tráfico cada 20 segundos de simulación se cambiará la media en la distribución utilizada. Por lo tanto, tendremos 480 segundos de tiempo de simulación para representar el tráfico de dos días.

En el siguiente ejemplo podemos apreciar lo explicado anteriormente y la figura 30, nos muestra el modelo de tráfico utilizado para nuestro estudio.

ejemplo:

De 3:00 a 4:00 horas tenemos 12601 llamadas, por lo tanto tenemos 6300.5 llamadas por hora, igual a 1.75 llamadas por segundo en dos horas. Para incrementar el tráfico: $(9 \times 1.75) = 15.75$ llamadas por segundo en dos horas. De este modo, en nuestro modelo en cada segundo de simulación se generaran 15.75 llamadas por segundo por un período de 20 segundos, en donde cada 20 segundos se cambiará el valor calculado (24 valores) construyendo un período de 480 segundos de simulación.

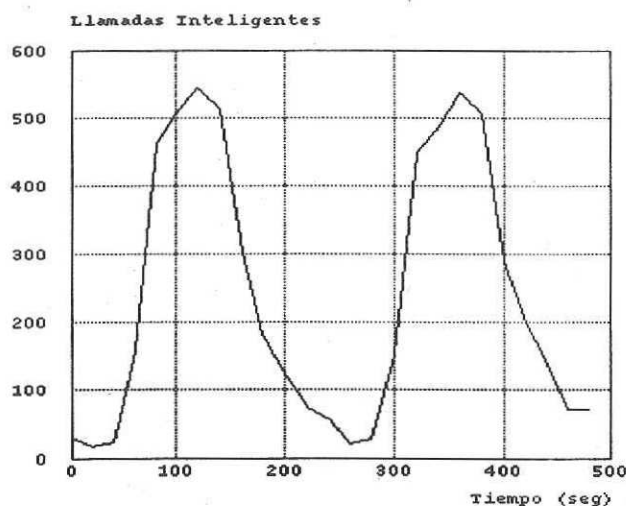


Figura 30 Tráfico bajo estudio.

Por otra parte, para simular la complejidad de los servicios en la red inteligente, el tráfico generado se divide de la siguiente manera: El 25% de las llamadas generadas no requieren la base de datos en el SCP, asumiendo que la información se encuentra en memoria principal, el 50% de las llamadas requieren de un acceso a la base de datos y el 25% restante requieren de dos accesos a la base de datos (Tabla IX).

Tabla IX Escenario de Llamadas Inteligentes.

<i>Número de accesos a la base de datos</i>	<i>Distribución (%)</i>
0	25
1	50
2	25

V.3 FASE I : SIN CONTROL

Para analizar nuestra configuración de red inteligente sin ningún mecanismo de control, utilizaremos el modelo de tráfico mostrado en la figura 30, el escenario de llamadas de la tabla IX, y los parámetros generales mostrados en la tabla III.

V.3.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP

En esta sección analizaremos en qué momento se llega a tener el problema de cuello de botella en la configuración del SCP, detectando la entidad que lo genera. El problema de cuello de botella es el principal factor que conforma un estado de congestión, éste se presenta cuando la cantidad de tráfico en espera de ser servido por un dispositivo sobrepasa la capacidad del mismo, ocasionando un incremento del buffer hasta posibles desbordamientos y con esto un incremento en el tiempo de respuesta.

La figura 31, nos muestra el número de llamadas inteligentes almacenadas en el buffer del CP en espera de ser atendidas con respecto al tiempo de simulación. Podemos ver que el buffer se incrementa en los periodos de 80-160 segundos y de 320-400 segundos de simulación. Comparando los puntos iniciales 80 y 320 segundos de simulación dentro de la figura del tráfico utilizado (Figura 30), nos podemos dar cuenta que la capacidad del sistema es aproximadamente de hasta 450 llamadas por segundo, donde cada procesador de llamada puede atender a 150 llamadas por segundo como límite. Pasando de esta cantidad el buffer del CP empieza a desbordarse y por lo tanto a descartar llamadas inteligentes enviadas al SCP en espera de ser atendidas cayendo en un estado de congestión.

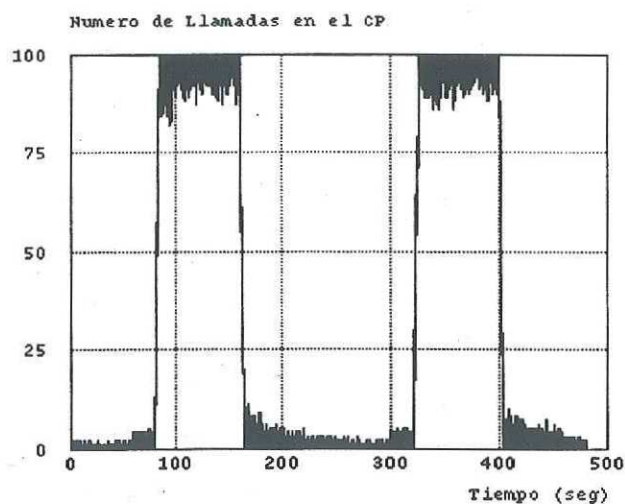


Figura 31 *Número de llamadas en el CP con respecto al tiempo de simulación.*

El número de llamadas que son almacenadas en el buffer del CP pueden ser llamadas que pueden o no requerir la base de datos según el escenario utilizado. Por lo tanto la capacidad del buffer en el CP es dividido en dos, en donde uno nos representa el número de llamadas inteligentes en espera de ser atendidas por el CPU y el otro el número de llamadas inteligentes en espera de ser servidas por la LDB. La figura 32, muestra el número de llamadas en espera de ser atendidas por el CPU en donde probablemente después podrán ser enviadas como respuestas al SSP o enviadas hacia la base de datos para tomar información necesaria para llevar a cabo el servicio inteligente. La figura 33, muestra el número de llamadas inteligentes en espera de ser atendidas por la base de datos y después enviadas hacia el CPU.

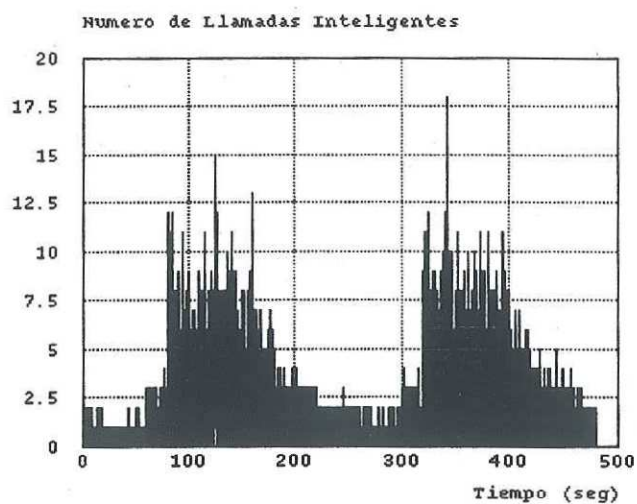


Figura 32 *Número de llamadas inteligentes en espera de ser atendidas por el CPU.*

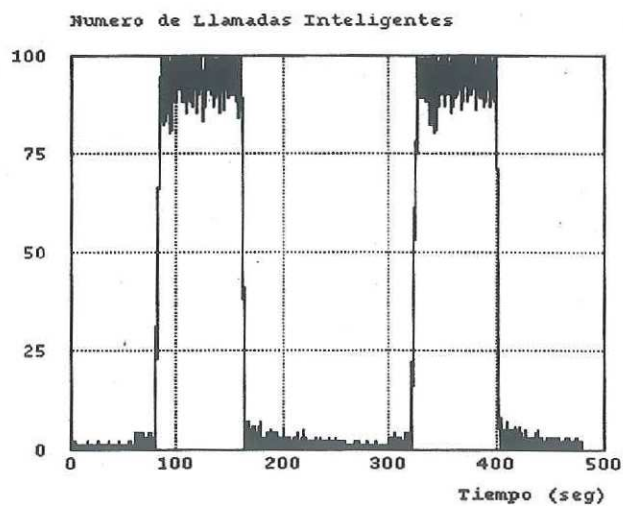


Figura 33 *Número de llamadas inteligentes en espera de ser atendidas por LDB.*

Las, figuras 34 y 35 muestran la utilización del CPU y la base de datos con respecto al tiempo de simulación.

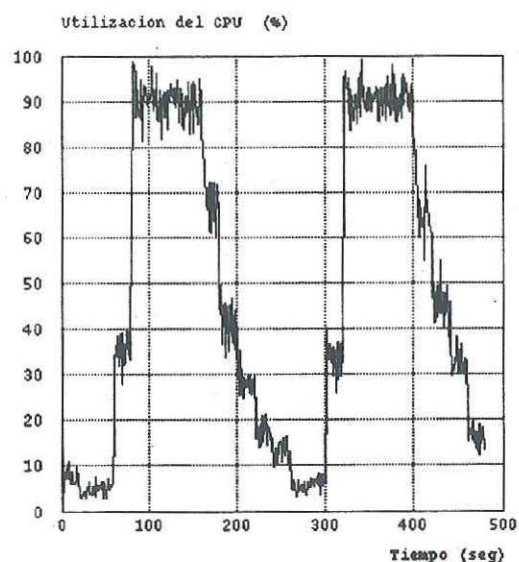


Figura 34 Utilización del CPU.

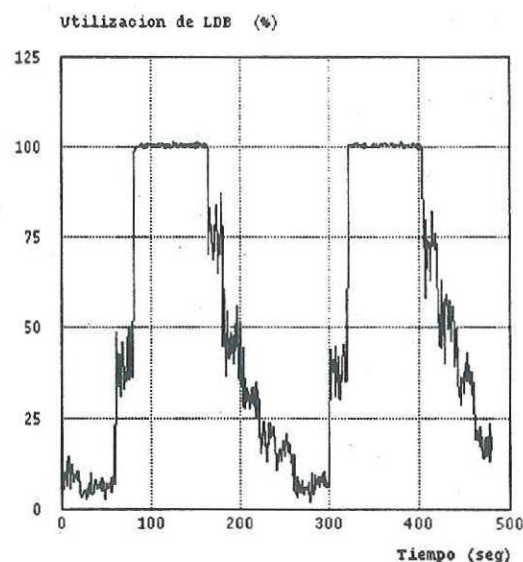


Figura 35 Utilización de LDB.

De este modo, analizando las figuras 32, 33, 34, y 35 podemos determinar el cuello de botella en el sistema. De las figuras podemos observar que en los periodos de alto tráfico (80-160 y 320-400 segundos) el CPU se mantiene en una media de 90% de utilización pudiendo atender el número de llamadas inteligentes que lo requieren, sin casi almacenar llamadas inteligentes en espera de ser atendidas. Por otra parte en este mismo período, la utilización en la base de datos se encuentra al 100% y almacenando un gran número de llamadas inteligentes en espera de ser atendidas, ocasionando con esto un estado de congestión debido al lento procesamiento. Por lo tanto, hemos encontrado que la base de datos local ocasiona el cuello de botella, resultando en que nuevos arribos al SCP sean descartados por el sobre flujo en el buffer del CP.

V.3.2 RETARDO DE LLAMADAS INTELIGENTES

La figura 36, muestra el tiempo de respuesta en forma dinámica con respecto al tiempo de cada una de las llamadas inteligentes enviadas hacia el SCP hasta que regresan al SSP con toda la información necesaria para llevar a cabo el servicio inteligente. En la red inteligente al momento en que la respuesta llega al SSP, el SSP prosigue a establecer la conexión necesaria entre el abonado que llama y al que se le llama a través de otros SSPs.

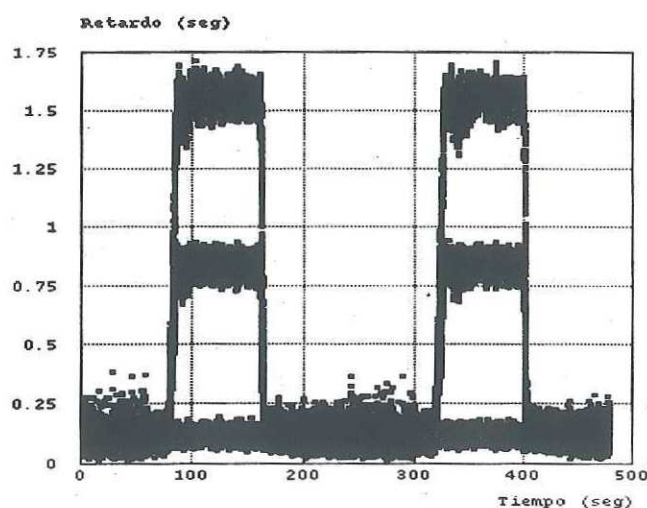


Figura 36 Retardo de Llamadas Inteligentes.

De la figura podemos observar que el tiempo de retardo aumenta en momentos de alto tráfico (80-160 y 320-400 segundos de tiempo de simulación). En donde claramente podemos notar que el retardo aumenta en periodos de posible estado de congestión. Del mismo modo podemos claramente identificar tres grupos de tiempo de retardo, esto es debido al escenario de llamadas inteligentes utilizado en donde podemos ver el retardo que sufren las llamadas que requieren de cero, uno y dos accesos a la base de datos. Los diferentes tipos de retardo pueden ser calculados de la siguiente manera:

En momentos en que la cantidad de llamadas generadas por los usuarios que llegan al SSP en donde el SSP envía de igual manera a cada CP de un SCP, no sobrepasa la capacidad del sistema (450 llamadas por segundo de simulación) y analizando el flujo de la llamada inteligente, por medio de la siguiente ecuación podemos calcular el retardo para una llamada inteligente que no requiere de acceder a la base de datos “R(0)” para ser servida.

$$R(0) = B_{ssp} * t_{ssp} + R_{ss7} + B_{cp} * t_{cpu} \quad (3)$$

En donde, B_{ssp} representa la cantidad de llamadas en el buffer del SSP en espera de ser atendidas (como el SSP envía todo arribo hacia el SCP en la configuración de no control, $B_{ssp} = 1$ llamada), t_{ssp} es el tiempo de servicio en el SSP (1.25ms.), R_{ss7} es el tiempo otorgado por la red de señalización número 7 por la transmisión de una llamada del SSP - SCP - SSP (100mspll²), B_{cp} representa el número de llamadas en el buffer del CP en espera de ser atendidas (en momentos que tráfico no sobre pasa la capacidad del sistema $B_{cp} = 1$ llamada y en momentos de alto tráfico $B_{cp} = 100$ llamadas) y t_{cpu} es el tiempo de procesamiento del CPU en el CP para una llamada inteligente (3.33ms.). De esta manera usando (3) podemos calcular el retardo que sufre una llamada inteligente que no requiere de acceder a la base de datos en el CP:

$$R(0) = (1)(1.25ms.) + 100mspll + (1)(3.33ms.) = 0.10458 spll^3.$$

² mspll = milisegundos por llamada.

³ spll = segundos por llamada.

Por lo tanto, 0.10458spll representa el valor de retardo promedio por llamadas que no requieren de acceder a la base de datos antes de que el tráfico sobrepase la capacidad del sistema. De la misma manera podemos calcular para las llamadas que requieren de uno "R(1)" y dos "R(2)" accesos a la base de datos antes del alto flujo de tráfico.

$$R(1) = B_{ssp} * t_{ssp} + R_{ss7} + 2 * B_{cp} * t_{cpu} + B_{cp} * t_{ldb} \quad (4)$$

$$R(2) = B_{ssp} * t_{ssp} + R_{ss7} + 3 * B_{cp} * t_{cpu} + 2 * B_{cp} * t_{ldb} \quad (5)$$

En donde t_{ldb} es el tiempo de servicio por la base de datos local. Debido a que la base de datos local representa el cuello de botella en el sistema y ésta es simulada por una distribución truncada con valor de 6ms-8ms-10ms, los cálculos se harán considerando la media de 8ms. De igual forma considerando que el número de llamadas generadas no sobrepasan la capacidad del sistema el valor de $B_{cp} = 1$ llamada. Entonces utilizando (4) y (5) tenemos:

Retardo de un acceso con $t_{ldb} = 8\text{ms.}$:

$$R(1) = (1)(1.25\text{ms}) + 100\text{mspll} + (2)(1)(3.33\text{ms}) + (1)(8\text{ms}) = 0.11591 \text{ spll.}$$

Por lo tanto 0.11591spll. es el retardo que sufre una llamada inteligente que requiere de un acceso a la base de datos en momentos de bajo tráfico. De la misma forma usando (5) calculamos el retardo de una llamada inteligente que requiere de dos accesos a la base de datos:

Retardo de dos accesos con $t_{ldb} = 8\text{ms.}$:

$$R(2) = (1)(1.25\text{ms}) + 100\text{mspll} + (3)(1)(3.33\text{ms}) + (2)(8\text{ms}) = 0.12724 \text{ spll.}$$

Así, 0.12724spll, es el retardo que sufre una llamada inteligente en acceder dos veces la base de datos en momentos de bajo tráfico.

Resumiendo:

$$R(0) = 0.10458 \text{ spll.}$$

$$R(1) = 0.11591 \text{ spll.}$$

$$R(2) = 0.12724 \text{ spll.}$$

$R(0)$, $R(1)$ y $R(2)$ son los retardos que sufre una llamada inteligente que requiere cero, uno o dos accesos a la base de datos en período de bajo tráfico.

De la misma forma podemos calcular el retardo que sufre una llamada inteligente en período de alto tráfico en donde la capacidad del sistema es alcanzada. En momentos de alto tráfico el buffer del CP está al límite almacenando llamadas que requieren de acceder a la base de datos, por lo tanto en momentos de alto tráfico $B_{cp} = 100$ llamadas y como el retardo es ocasionado por el cuello de botella debido al lento procesamiento de servicio de la base de datos, (3) , (4) y (5) se pueden reescribir de la siguiente manera:

$$R(0) = B_{ssp} * t_{ssp} + R_{ss7} + t_{cpu} \quad (6)$$

$$R(1) = B_{ssp} * t_{ssp} + R_{ss7} + 2 * t_{cpu} + B_{cp} * t_{ldb} \quad (7)$$

$$R(2) = B_{ssp} * t_{ssp} + R_{ss7} + 3 * t_{cpu} + 2 * B_{cp} * t_{ldb} \quad (8)$$

La diferencia con (3), (4) y (5) es que el tiempo de procesamiento del CPU en el CP (t_{cpu}) no es multiplicado por la cantidad de llamadas en el buffer del CP (B_{cp}) debido a que este no es el dispositivo que nos ocasiona el cuello de botella en el sistema. Por lo tanto utilizando (6), (7) y (8) podemos calcular lo siguiente:

Retardo de llamadas con cero accesos en momentos en que el tráfico sobrepasa la capacidad del sistema:

$$R(0) = (1)(1.25\text{ms.}) + 100\text{mspll} + 3.33\text{mspll.} = 0.10458\text{spll.}$$

De la misma manera se calcula el retardo para llamadas que requieren de un acceso a la base de datos para poder completar el servicio:

Retardo de un acceso con $t_{ldb} = 8\text{ms.}$

$$R(1) = (1)(1.25\text{ms.}) + 100\text{mspll} + (2)(3.33\text{mspll.}) + (100)(8\text{ms.}) = 0.9079\text{spll.}$$

En donde 0.9079spll. es el retardo que sufre una llamada inteligente que requiere un acceso a la base de datos en momento de alto tráfico. De la misma forma tenemos que para:

Retardo de dos accesos con $t_{ldb} = 8\text{ms.}$

$$R(2) = (1)(1.25\text{ms.}) + 100\text{mspll} + (3)(3.33\text{mspll.}) + (2)(100)(8\text{ms.}) = 1.71124\text{spll.}$$

En donde 1.7112spll. es el retardo que sufre una llamada inteligente que requiere de dos accesos a la base de datos.

Resumiendo:

$$R(0) = 0.10458\text{spll.}$$

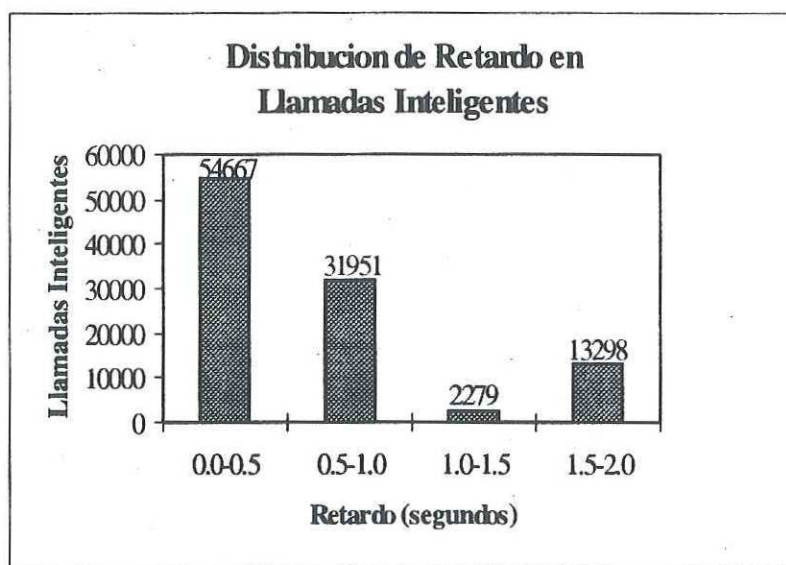
$$R(1) = 0.9079\text{spll.}$$

$$R(2) = 1.71124\text{spll.}$$

$R(0)$, $R(1)$ y $R(2)$ son los retardos que sufre una llamada inteligente que requiere de cero, uno o dos accesos a la base de datos en período de alto tráfico, respectivamente.

De esta manera, regresando a la figura 36, claramente podemos identificar los tres diferentes retardos calculados en los periodos de alto tráfico. Cabe aclarar que estos resultados son de acuerdo al escenario de llamadas inteligentes, en donde se está considerando que el tipo de servicio inteligente es muy complejo, el uso de diferentes escenario de llamadas en el sistema sin control puede ser visto en (Ouellette y Barbosa, 1997).

En la figura 37, podemos ver un histograma que muestra la distribución de retardo en las llamadas inteligentes. En donde el requerimiento principal es de que al menos el 95% de las llamadas previamente admitidas por el sistema y enviadas al SCP sean servidas satisfactoriamente dentro del límite de tiempo requerido para cumplir con la calidad de servicio inteligente (2 segundos). El número de llamadas marcadas como pérdidas, se refiere a las llamadas que fueron previamente aceptadas y enviadas al SCP por el SSP en espera de ser servidas y fueron descartadas por el sobreflujo en el buffer del CP en periodos de alto tráfico o porque sobrepasaron el límite de tiempo requerido no cumpliendo con la calidad de servicio que una llamada inteligente requiere.



#Llamadas Pérdidas = 15252

Figura 37 Distribución de retardo de llamadas Inteligentes.

De la misma figura decimos que el 13% de las llamadas admitidas y enviadas hacia el SCP no cumplieron con la calidad de servicio requerida por una llamada inteligente o fueron descartadas por momentos de congestión. Por otra parte se cumplió con el 87% de las llamadas, siendo este un valor insuficiente para considerar el desempeño del sistema como adecuado.

V.3.3 GOODPUT EN EL SCP

En esta sección estudiaremos el Goodput del sistema en el SCP. En donde el GOODPUT en nuestro estudio es un parámetro que expresa el efecto del cuello de botella en el SCP y por lo tanto refleja la eficiencia en el uso del SCP. El Goodput está representado por la suma de llamadas exitosamente servidas por el SCP entre la suma del número total de llamadas enviadas al SCP en espera de ser servidas por el SSP. (Berger,

1991). Por lo tanto, el Goodput en nuestro estudio está representado por:

$$\text{Goodput (\%)} = \frac{\sum_{t=1}^{t_{\text{simulación}}} (\text{llamadas exitosamente servidas por el SCP})}{\sum_{t=1}^{t_{\text{simulación}}} (\text{llamadas enviadas al SCP por SSPs})} * 100 \quad (9)$$

De esta manera con (9) podemos observar el efecto del cuello de botella en el sistema en forma dinámica con respecto al tiempo de simulación. En donde si el Goodput permanece estable muy cercano a 100% nos indica que el sistema está fuera de un posible estado de congestión, representando con esto que el caudal eficaz en el sistema es adecuado y que todas las llamadas son servidas satisfactoriamente. De otra forma, cuando el Goodput del sistema decrece, podemos asegurar que es debido a un posible estado de congestión en donde el número de llamadas entrantes al SCP no son atendidas satisfactoriamente, decrementando con esto el caudal eficaz del sistema e incrementando el tiempo de servicio a una llamada inteligente debido a un posible estado de congestión.

La figura 38, muestra el Goodput en uno de los CP que conforma al SCP. En donde podemos ver que al inicio de alto tráfico, a 80 segundos del tiempo de simulación, el Goodput en el sistema decrece considerablemente indicando con esto un posible estado de congestión y un decremento en el caudal eficaz en el SCP. Podemos también ver que el Goodput no se normaliza después de los periodos de alto tráfico, debido a que estudiaremos el Goodput del sistema en forma acumulativa con el tiempo, con el fin de obtener mas información para comparar de los resultados usando los mecanismos de control de flujo bajo estudio.

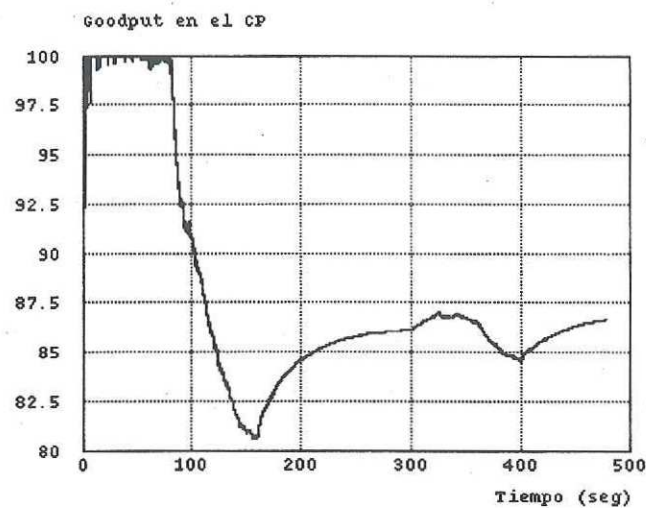


Figura 38 *Goodput en el CP.*

V.4 FASE I : CON CONTROL DE LA VENTANA (WINDOW)

Para analizar nuestra configuración de red inteligente implementando el mecanismo de control de flujo de la Ventana, utilizaremos el modelo de tráfico mostrado en la figura 30, el escenario de llamadas de la tabla IX, los parámetros generales mostrados en la tabla III, y los parámetros óptimos encontrados para utilizar el control de la Ventana dados en la tabla VII.

V.4.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP

La figura 39, muestra el número de llamadas inteligentes que se almacenan en el buffer del CP en espera de ser atendidas con respecto al tiempo de simulación. El mecanismo de control de flujo de la Ventana mantiene el tamaño de su ventana máxima (WinMax) en periodos de alto tráfico, igual al tamaño máximo del buffer en el CP, con esto

manteniendo al SCP fuera de posibles estados de congestión y así cumplir con la calidad de servicio que una llamada inteligente requiere. Así en nuestro estudio, el SSP no puede tener mas de 100 llamadas fuera en espera de ser atendidas por cada CP que conforma el SCP.

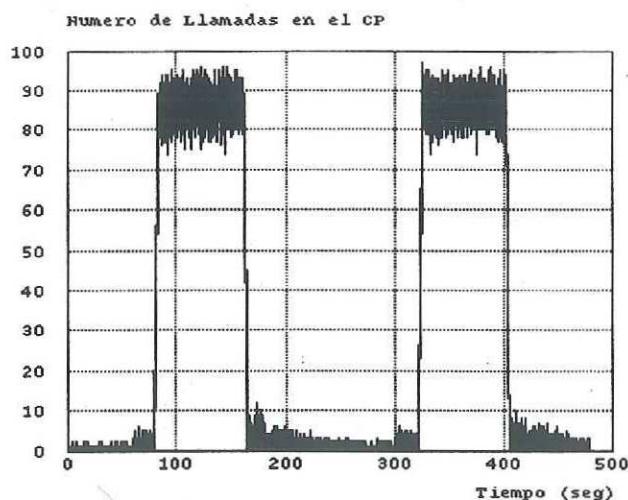


Figura 39 *Número de Llamadas en el CP con respecto al tiempo de simulación utilizando el control de la Ventana.*

De la figura podemos notar, que en los periodos de 80-160 y de 320-400 segundos de tiempo de simulación en donde el tráfico generado empieza a sobrepasar la capacidad del sistema, el buffer en el CP se ve incrementado al 90%. Lo anterior es debido a que el mecanismo bajo estudio aproximadamente a 83 y 323 segundos de tiempo de simulación monitorea que tiene mas de 100 llamadas fuera en espera de ser servidas, en este momento el SSP cierra su ventana parando de transmitir llamadas inteligentes hacia el SCP y empezando almacenar en su buffer local hasta un máximo de 200 llamadas. Sin embargo, debido al funcionamiento del mecanismo el SSP abre su ventana cada vez que este recibe una respuesta por el SCP permitiendo con esto enviar una nueva llamada inteligente hacia el

SCP . Por otra parte nos damos cuenta de que al hacer uso del mecanismo de control de la Ventana el buffer incrementa hasta un máximo de 90 llamadas por segundo en periodos de alto tráfico, considerando que las 10 llamadas por segundo restante que monitorea el SSP se encuentran en la red de señalización en espera de ser enviadas hacia el SCP o hacia el SSP.

Debido al escenario de llamadas inteligentes utilizado, es muy probable que en periodos de alto tráfico las llamadas que se encuentran almacenadas en el SCP sean aquellas que requieren el uso de la base de datos. Entonces el buffer del CP está formado por llamadas en espera de ser atendidas tanto por el CPU como por la LDB. Para verificar lo anteriormente mencionado las figuras 40, y 41, muestran el número de llamadas inteligentes en espera de ser servidas por el CPU y por la base de datos en el CP, respectivamente.

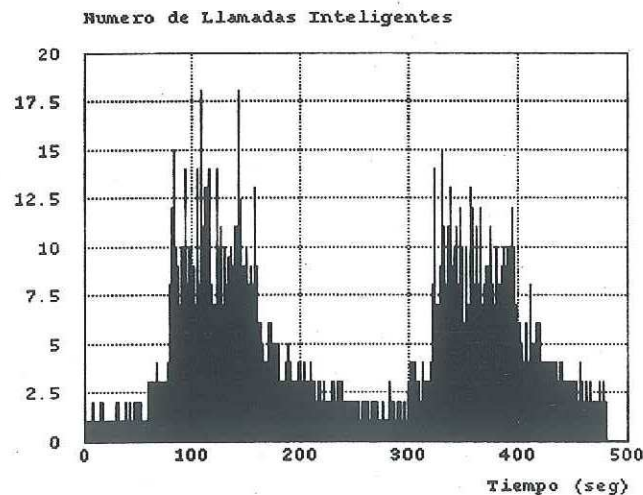


Figura 40 *Número de llamadas inteligentes en espera de ser atendidas por el CPU utilizando el mecanismo de la Ventana.*

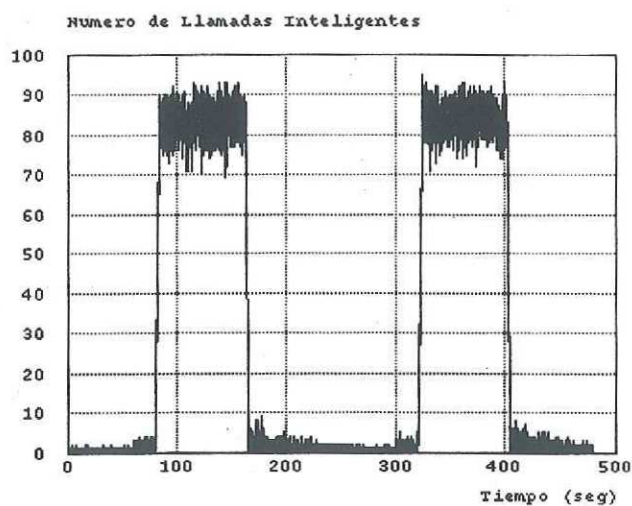


Figura 41 *Número de llamadas inteligentes en espera de ser atendidas por LDB utilizando el mecanismo de la Ventana.*

De la misma forma, las figuras 42, y 43, muestran la utilización con respecto al tiempo del CPU y la base de datos usando el mecanismo de control de la Ventana, con el fin de determinar cual es el dispositivo que nos proporciona el problema de cuello de botella en el sistema, con esto incrementando el retardo de una llamada inteligente.

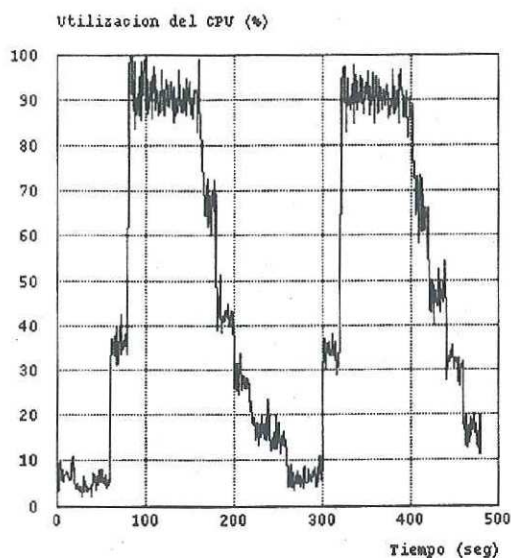


Figura 42 Utilización del CPU.

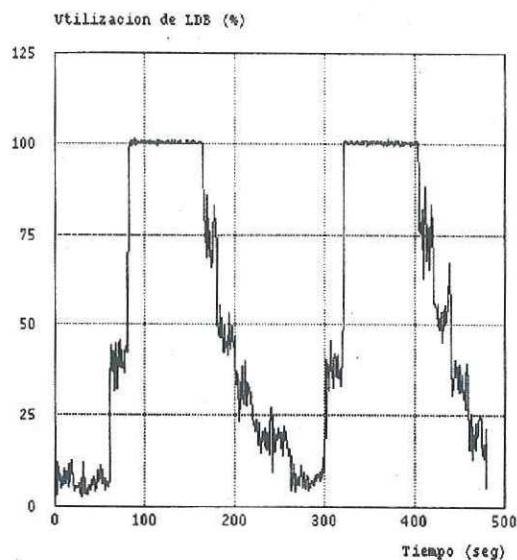


Figura 43 Utilización de LDB.

De este modo, analizando las figuras 40, 41, 42 y 43 nos podemos dar cuenta de que en el período de alto tráfico el buffer del CPU se mantiene aproximadamente a una media del 15% de su capacidad, y manteniendo un 90% de utilización, almacenando llamadas que deben de ser procesadas para después mandarse como respuesta al SSP o identificar llamadas que requieren de acceder a la base de datos y por lo tanto enviarse a la base de datos. Por otra parte el buffer de la base de datos se mantiene en una media del 85% con una utilización del 100%, entonces la base de datos es la entidad que nos proporciona el problema de cuello de botella en el sistema, debido al lento procesamiento para servir a las llamadas inteligentes, provocando con esto un almacenamiento masivo y un incremento en el tiempo de respuesta. Gracias al mecanismo de control de flujo de la Ventana el buffer en el CP no se ve congestionado, por lo tanto este mecanismo asegura que el SCP este fuera de posibles estados de congestión y que la probabilidad de bloqueo en el SCP se decremente

considerablemente.

V.4.2 RETARDO DE LLAMADAS INTELIGENTES

La figura 44, muestra el tiempo de respuesta en forma dinámica con respecto al tiempo de simulación, de cada una de las llamadas inteligentes enviadas al SCP, hasta que el SSP recibe la respuesta con toda la información necesaria para completar el servicio inteligente, utilizando el mecanismo de control de flujo de la Ventana.

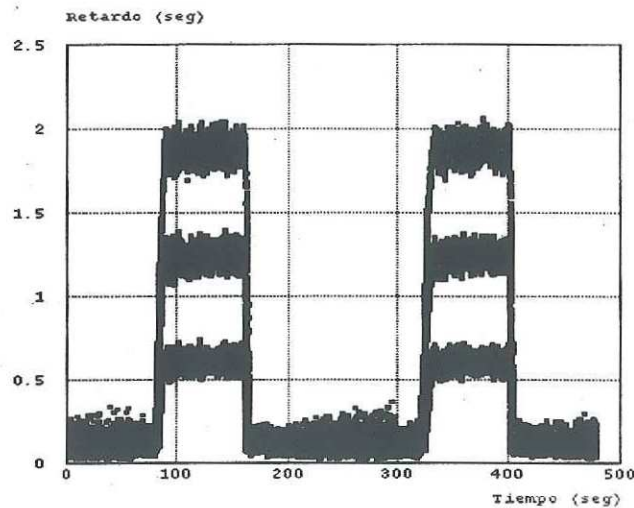


Figura 44 Retardo de Llamadas Inteligentes con control de la Ventana.

De la figura podemos observar que el tiempo de retardo aumenta en momentos de alto tráfico (80-160 y 320-400 segundos de tiempo de simulación). Claramente de la figura podemos ver que el retardo aumenta en momentos que el buffer del CP aumenta y en este mismo tiempo el SSP cierra su ventana de transmisión debido al monitoreo de 100 llamadas fuera del sistema en espera de ser atendidas. En momentos en que el SSP cierra su ventana, éste empieza a almacenar llamadas inteligentes en su buffer local hasta un máximo de 200

llamadas. Si el buffer del SSP llega a su límite, éste empieza a bloquear llamadas avisando rápidamente al abonado de la falta de recursos en el sistema (típicamente el abonado es avisado por medio de un tono de ocupado).

De la misma figura podemos notar que el retardo es mayor en comparación al retardo obtenido en la configuración sin control, esto es debido a que en el momento en que el SSP cierra su ventana de transmisión este empieza a almacenar llamadas inteligentes las cuales son afectadas por un retardo adicional. En donde este retardo adicional representa el tiempo en que el SSP recibe una respuesta por el SCP, permitiendo con esto abrir temporalmente su ventana y enviar la siguiente llamada inteligente hacia el SCP. Como nos dimos cuenta en la sección anterior, en donde en momentos de alto tráfico el CP almacena llamadas que requieren de acceder a la base de datos, por lo tanto el retardo extra puede ser representado por el tiempo que espera una llamada inteligente en el CP a ser procesada por ultima vez por el CPU para después enviarse como respuesta al SSP y este proceda con el envío de una nueva llamada inteligente. Entonces en momentos de alto tráfico el CP tiene almacenado una media de 85 llamadas en espera de ser atendidas por la base de datos y después posiblemente procesadas por el CPU por última vez. Considerando lo anterior el retardo adicional puede estar expresado como:

$$R_e = \text{Retardo adicional} = B_{cp} * t_{cpu} \quad (10)$$

Considerando $B_{cp} = 85$ llamadas y $t_{cpu} = 3.33\text{ms.}$ y utilizando (10) tenemos:

$$R_e = (85 \text{ llamadas}) (3.33\text{ms.}) = 0.28305\text{spll.}$$

En donde 0.28305spll. es el tiempo medio extra que debe de esperar una llamada inteligente en el SSP por una respuesta del SCP solo en momentos de alto tráfico.

El retardo medio que sufre una llamada inteligente en momentos en que el tráfico generado no sobrepasa la capacidad del sistema es el mismo calculado en el caso de sin control, puesto que el sistema puede procesar las llamadas inteligentes sin ninguna complicación. Por lo tanto tenemos que :

$$R(0) = 0.10458 \text{ spll.}$$

$$R(1) = 0.11591 \text{ spll.}$$

$$R(2) = 0.12724 \text{ spll.}$$

En donde $R(0)$, $R(1)$ y $R(2)$, son los retardos calculados para una llamada inteligente que requiere cero, uno y dos accesos a la base de datos, respectivamente.

De la misma forma podemos calcular el retardo que sufre una llamada inteligente en período de alto tráfico en donde la capacidad del sistema es alcanzada usando el mecanismo de control de la Ventana. Entonces hacemos uso de (6), (7) y (8) reescribiéndolas de la siguiente manera:

$$R(0) = B_{ssp} * t_{ssp} + R_{ss7} + t_{cpu} + R_e \quad (11)$$

$$R(1) = B_{ssp} * t_{ssp} + R_{ss7} + 2 * t_{cpu} + B_{cp} * t_{ldb} + R_e \quad (12)$$

$$R(2) = B_{ssp} * t_{ssp} + R_{ss7} + 3 * t_{cpu} + 2 * B_{cp} * t_{ldb} + R_e \quad (13)$$

La diferencia con (6), (7) y (8) es que los tiempos de respuesta de una llamada inteligente que requiere de cero, uno y dos accesos a la base de datos se ven incrementados por un retardo adicional (R_e) debido al funcionamiento del mecanismo de control de flujo de la Ventana explicado anteriormente. De esta manera utilizando (11), (12), (13) y considerando que en momentos de alto tráfico el buffer del CP mantiene una media de 85 llamadas ($B_{cp} = 85$), de este mismo modo cuando el SSP cierra su ventana de transmisión

empieza a almacenar llamadas llegando hasta 200 llamadas ($B_{sp} = 200$), podemos calcular lo siguiente:

Retardo de llamadas con cero accesos en momentos en que el tráfico sobrepasa la capacidad del sistema utilizando el control de la Ventana con (11) tenemos:

$$R(0) = (200)(1.25\text{ms.}) + 100\text{mspll} + 3.33\text{mspll} + 0.28305\text{spll} = 0.63638\text{spll}.$$

De la misma manera se calcula el retardo para llamadas que requieren de un acceso a la base de datos para poder a completar el servicio utilizando el control de la Ventana con (12) tenemos:

Retardo de un acceso con $t_{db} = 8\text{ms.}$

$$R(1) = (200)(1.25\text{ms}) + 100\text{mspll} + (2)(3.33\text{mspll}) + (85)(8\text{ms}) + 0.28305\text{spll} = 1.31971\text{spll}.$$

En donde 1.31971spll. es el retardo que sufre una llamada inteligente que requiere un acceso a la base de datos en momento de alto tráfico utilizando el mecanismo de control de flujo de la Ventana. De la misma forma tenemos (13) tenemos :

Retardo de dos accesos con $t_{db} = 8\text{ms.}$

$$R(2) = (200)(1.25\text{ms}) + 100\text{mspll} + (3)(3.33\text{mspll}) + (2)(85)(8\text{ms}) + 0.28305\text{spll} = 2.003\text{spll}.$$

En donde 2.003spll. es el retardo que sufre una llamada inteligente que requiere de dos accesos a la base de datos utilizando el mecanismo de control de la Ventana en período de alto tráfico.

Resumiendo:

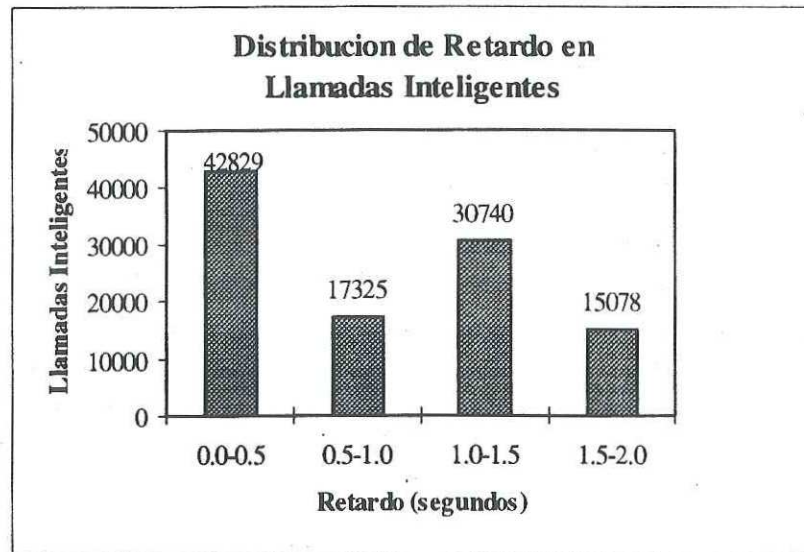
$$R(0) = 0.63638\text{spll.}$$

$$R(1) = 1.31971\text{spll.}$$

$$R(2) = 2.003\text{spll.}$$

En donde $R(0)$, $R(1)$ y $R(2)$ son los retardos que sufre una llamada inteligente en un período de alto tráfico, que requieren de cero, uno o dos accesos a la base de datos respectivamente, con el fin de llevar a cabo un servicio inteligente satisfactoriamente. De esta manera, regresando a la figura 44, claramente podemos identificar los tres diferentes retardos calculados en los periodos de alto tráfico usando el mecanismo de control de flujo de la Ventana. Pudiendo ver claramente los tres grupos de llamadas que conforman al escenario de llamada inteligente utilizado.

En la figura 45, podemos ver un histograma que muestra la distribución de retardo en las llamadas inteligentes. En donde el requerimiento principal es de que al menos el 95% de las llamadas previamente admitidas por el sistema y enviadas al SCP sean servidas satisfactoriamente dentro del límite de tiempo requerido para cumplir con la calidad de servicio inteligente (2 segundos). El número de llamadas marcadas como bloqueadas, se refieren aquellas llamadas que no fueron admitidas por el SSP debido a la falta de recursos en la red y que rápidamente el abonado pudo ser avisado.



#Llamadas Pérdidas = 0 #Llamadas Bloqueadas = 11526

Figura 45 Distribución de retardo de llamadas Inteligentes con control de la Ventana.

Del histograma podemos ver que el 10% de las llamadas generadas por los usuarios en el tiempo de simulación fueron básicamente bloqueadas y que el 100% de las llamadas admitidas por el sistema fueron atendidas satisfactoriamente. Por lo tanto, el desempeño del mecanismo se considera adecuado en nuestro estudio. Sin embargo, se buscará que la probabilidad de bloqueo en el SSP disminuya conservando la calidad de servicio obtenida.

V.4.3 GOODPUT EN EL SCP

En esta sección analizaremos el Goodput del sistema de la misma forma que lo hemos estado haciendo. Sin embargo, ahora estudiaremos el Goodput del sistema utilizando el mecanismo de control de flujo de la Ventana.

La figura 46, nos presenta el Goodput monitoreado en el transcurso de la simulación con respecto al tiempo de un CP en el SCP utilizando el mecanismo de control bajo estudio.

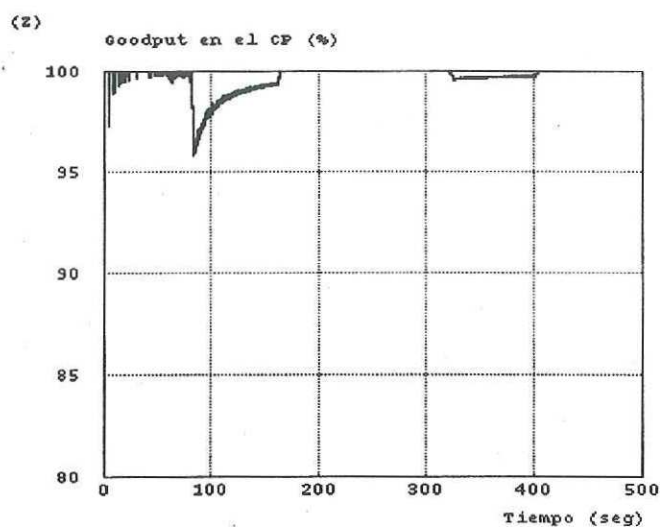


Figura 46 Goodput en el CP con control de la Ventana.

De la figura podemos ver que al inicio de periodos de alto tráfico el Goodput en el CP se decrementa (80 y 320 segundos tiempo de simulación), esto debido a que un número de llamadas inteligentes empiezan a ser almacenadas en el buffer del CP en espera de ser servidas por la base de datos, quien ocasiona el cuello de botella en el sistema. En estos momentos el SSP cierra su ventana de transmisión en donde espera una respuesta por el SCP para abrirla momentáneamente. De este modo este mecanismo de control mantiene un caudal eficaz estable en el SCP en momentos de alto tráfico. Permitiendo con esto, que el Goodput se normalice rápidamente y vuelva a llegar aproximadamente al 100%.

Por lo tanto, concluimos que el uso del mecanismo de control de flujo de la Ventana, mantiene un caudal eficaz adecuado en el SCP en momentos de alto tráfico, permitiendo con esto que toda llamada inteligente sea atendida y enviada al SSP, manteniendo un Goodput al 100% y dejando al SCP fuera de posibles estados de congestión.

V.5 FASE I : CON CALL GAPPING

Para analizar nuestra configuración de red inteligente implementando el mecanismo de control de flujo Call Gapping, utilizaremos el modelo de tráfico mostrado en la figura 30, el escenario de llamadas de la tabla IX, los parámetros generales mostrados en la tabla III, y los parámetros óptimos encontrados para utilizar Call Gapping mostrados en la tabla IV.

V.5.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP

La figura 47, muestra el número de llamadas inteligentes almacenadas en el buffer del CP con respecto al tiempo de simulación. Recordando que el mecanismo de control de flujo Call Gapping está implementado en cada CP del SCP, en donde al inicio de un intervalo de tiempo dado, el CP monitorea el estado del buffer comparándolo con el valor de dos umbrales en donde, si el umbral superior es excedido el control es activado empezando el CP a elaborar estadísticas de tiempo de servicio de llamadas y a crear un mensaje de control para ajustar la tasa de salidas en el SSP.

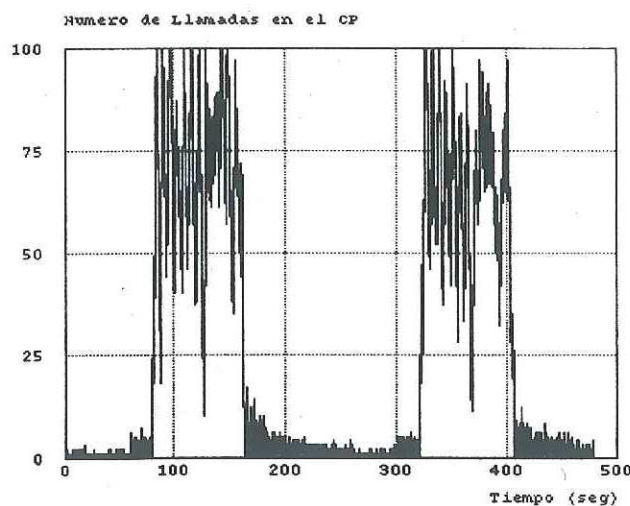


Figura 47 *Número de llamadas en el CP con respecto al tiempo de simulación utilizando el mecanismo de control Call Gapping.*

Considerando el funcionamiento de Call Gapping, podemos entonces de la figura analizar que en momentos de bajo tráfico, el número de llamadas que se monitorean del buffer en el CP al inicio de cada intervalo de muestreo es inferior al umbral superior, por lo tanto el mecanismo permanece desactivado. Sin embargo en momentos en que el tráfico empieza a llegar a la capacidad del sistema (80-160 y 320-400 segundos de tiempo de simulación) y en donde el intervalo de muestreo aun no se cumple, se tiene la probabilidad de que el CP caiga en estado de congestión empezando a descartar llamadas inteligentes que se enviaron en espera de ser servidas. Por otra parte en el momento en que el intervalo de muestreo inicia, se da cuenta de que el número de llamadas sobrepasa el umbral superior que protege al CP de un posible estado de congestión, empezando a elaborar estadísticas y a crear un mensaje de control que es enviado al SSP a través de la red de señalización para que ajuste su tasa de salidas, sin embargo, el tiempo que transcurre en que el mensaje de

control llegue al SSP el SCP puede estar en estado de congestión. En el momento que el SSP ajusta su tasa de salida, el número de llamadas almacenadas en el CP empieza a disminuir, esperando otro intervalo de muestreo el cual se de cuenta que el número de llamadas en el CP es menor al umbral inferior, en estos momentos el CP vuelve a procesar estadísticas y a crear un mensaje de control para normalizar la tasa de salidas en el SSP. Y así sucesivamente durante los periodos de alto tráfico en donde el mecanismo es activado. De esta forma, podemos concluir que el mecanismo de Call Gapping cambia el caudal en el SCP constantemente, tal y como lo muestra la figura.

Las figuras 48, y 49, muestran el número de llamadas que conforman el total de llamadas almacenadas en el buffer del CP, en donde podemos ver el número de llamadas en espera de ser atendidas por el CPU, para después enviarse al SSP o a la base de datos y el número de llamadas en espera de ser atendidas por la base de datos, para después enviarse al CPU, respectivamente.

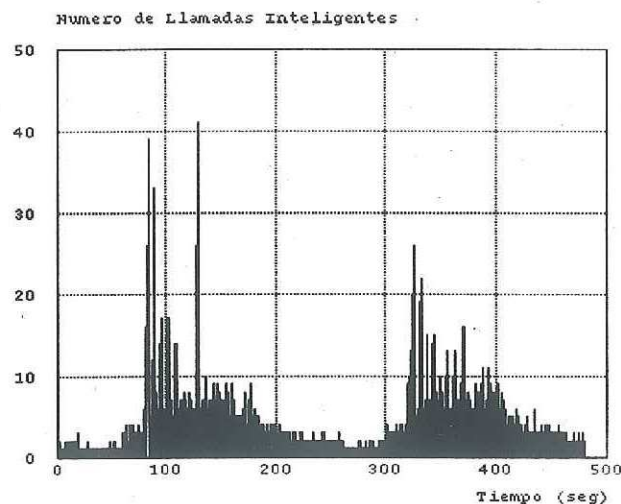


Figura 48 *Número de llamadas en espera de ser atendidas por el CPU con Call Gapping.*

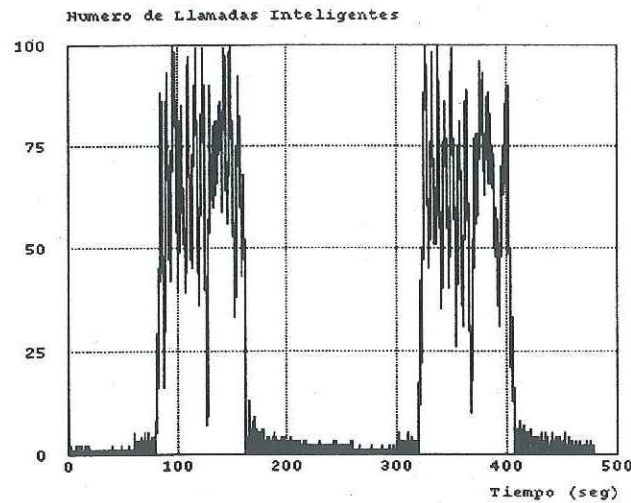


Figura 49 Número de llamadas en espera de ser atendidas por LDB con Call Gapping.

Las figuras 50, y 51, muestran respectivamente las utilizaciones correspondientes al uso del CPU y la base de datos con respecto al tiempo de simulación con el uso de Call Gapping, con el fin de detectar el problema del cuello de botella en el sistema.

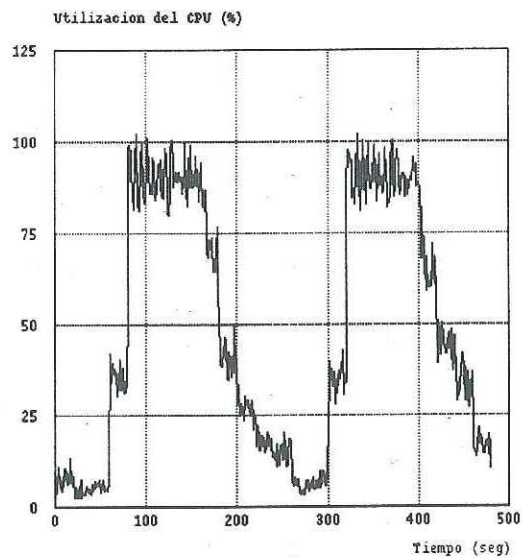


Figura 50 Utilización del CPU

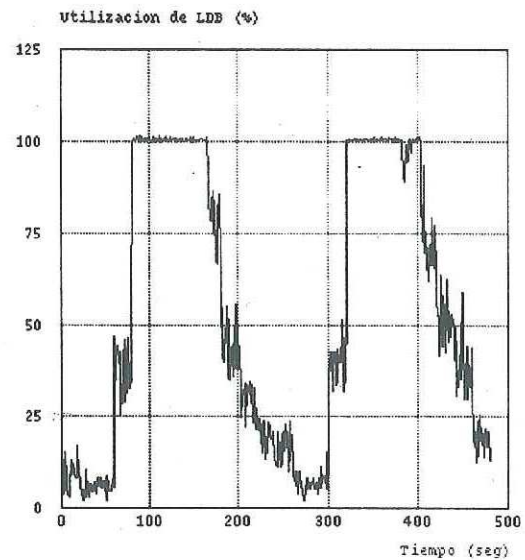


Figura 51 Utilización de LDB.

Analizando las figuras 48, 49, 50 y 51, podemos detectar el problema de cuello de botella en el sistema, el cual ocasiona los posibles estados de congestión y el incremento del tiempo de respuesta a una llamada inteligente. Entonces, de las figuras podemos notar que el número de llamadas almacenadas en espera de ser atendidas por el CPU aumenta hasta un 25% de la capacidad máxima del buffer en el CP, manteniendo una media de 90% de utilización, esto debido al comportamiento del mecanismo de control, puesto que en momentos de alto tráfico el CPU esta ocupado llevando a cabo el trabajo del mecanismo de control sin atender llamadas inteligentes, permitiendo con esto almacenar más llamadas en un momento dado. Sin embargo, el número de llamadas almacenadas no es demasiado para que el CPU se considere la entidad que nos proporcione el cuello de botella. De otro modo, el número de llamadas inteligentes almacenadas en espera de ser servidas por la base de datos es de una media de 75% de llamadas y la utilización de la base de datos está a un 100%, por lo tanto, de nuevo encontramos que la base de datos es la entidad que ocasiona el problema de cuello de botella en el sistema debido a su lento servicio de procesamiento. Sin embargo, si el período de monitoreo fuera mas corto al encontrado óptimo, el CPU se convertiría en la entidad que nos proporciona el problema de cuello de botella en el SCP, debido a que estaría mas tiempo el CPU procesando estadísticas y elaborando mensajes de control, permitiendo que el número de llamadas almacenadas en el buffer incremente y caiga el SCP en posibles estados de congestión.

V.5.2 RETARDO DE LLAMADAS INTELIGENTES

La figura 52, muestra el tiempo de respuesta en forma dinámica con respecto al tiempo de simulación de cada una de las llamadas inteligentes enviadas al SCP, hasta que el SSP recibe la respuesta con toda la información necesaria para completar el servicio inteligente, utilizando el mecanismo de control de flujo Call Gapping.

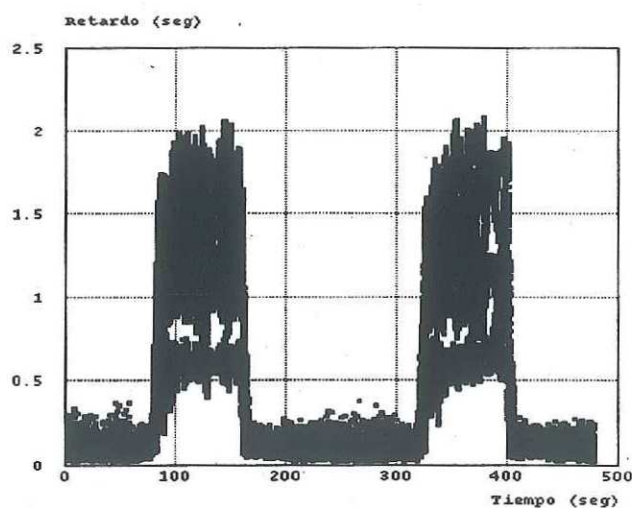


Figura 52 Retardo de Llamadas Inteligentes utilizando el mecanismo de control Call Gapping.

De la figura podemos observar que el tiempo de retardo aumenta en momentos de alto tráfico (80-160 y 320-400 segundos de tiempo de simulación), del mismo modo en los periodos en que el buffer del CP en el SCP aumenta. También nos damos cuenta de que el retardo se incrementa a comparación del caso de no control, debido al uso del mecanismo de control de flujo Call Gapping, en donde el mecanismo hace uso de periodos de monitoreo del buffer, activando o desactivando el control. En momentos en los que el control es activado, el CP crea un mensaje de control para ser enviado al SSP, en donde el SSP al recibir el mensaje adquiere un valor de bandas de llamadas el cual ajusta la tasa de salidas

de llamadas enviadas al SCP. En momentos en que el SSP ajusta su tasa de salida, éste empieza a almacenar llamadas en su buffer local en espera de ser atendidas, por lo tanto estas llamadas almacenadas son afectadas por un incremento en su tiempo de servicio.

Por otra parte, en la figura no podemos identificar claramente tres grupos de retardos de respuesta para una llamada inteligente que requiere de cero, uno o dos accesos a la base de datos, como lo hacíamos anteriormente. Esto debido a que Call Gapping cambia constantemente la tasa de salidas en el SSP, aproximadamente cada 0.25 segundos, los cuales representan el período de muestreo utilizado por Call Gapping en nuestro estudio, y del mismo modo en periodos de alto tráfico, este valor representa el retardo extra que sufre una llamada inteligente en espera de ser atendida y almacenada en el SSP. Entonces el retardo adicional es representado como:

$$R_e = \text{Valor de período de muestreo en Call Gapping} = 0.25\text{spll.} \quad (14)$$

Cuando el tráfico generado no sobrepasa la capacidad del sistema, el retardo medio que sufre una llamada inteligente es el mismo calculado en el caso de sin control, puesto que el sistema puede procesar las llamadas inteligentes sin ninguna complicación. Por lo tanto tenemos que :

$$R(0) = 0.10458\text{spll.}$$

$$R(1) = 0.11591\text{spll.}$$

$$R(2) = 0.12724\text{spll.}$$

En donde $R(0)$, $R(1)$ y $R(2)$, son los retardos calculados para una llamada inteligente que requiere cero, uno y dos accesos a la base de datos, respectivamente.

De la misma forma podemos calcular el retardo que sufre una llamada inteligente en periodos de alto tráfico, requiriendo cero, una o dos veces a la base de datos. Por medio de las figuras que muestran el comportamiento del buffer en el CP, podemos calcular una media de llamadas en el buffer en periodos de alto tráfico y monitoreando el buffer en el SSP de igual manera podemos calcular una media del número de llamadas almacenadas. Por lo tanto consideremos los valores siguientes para llevar a cabo los cálculos: $B_{cp} = 75$ llamadas, $B_{ssp} = 175$ llamadas. Utilizando (11), (12) y (13) podemos calcular lo siguiente:

Utilizando (11) calculamos el retardo de llamadas que no requieren de la base de datos en momentos en que el tráfico sobrepasa la capacidad del sistema utilizando Call Gapping :

$$R(0) = (175)(1.25\text{ms.}) + 100\text{mspll} + 3.33\text{mspll.} + 0.25\text{spll.} = 0.57208\text{spll.}$$

Utilizando (12) de la misma manera se calcula el retardo para llamadas que requieren de un acceso a la base de datos para poder a completar el servicio utilizando Call Gapping:

Retardo de un acceso con $t_{db} = 8\text{ms.}$

$$R(1) = (175)(1.25\text{ms.}) + 100\text{mspll} + (2)(3.33\text{mspll.}) + (75)(8\text{ms.}) + 0.25\text{spll} = 1.17541\text{spll.}$$

En donde 1.17541spll. es el retardo que sufre una llamada inteligente que requiere un acceso a la base de datos en momento de alto tráfico utilizando el mecanismo de control de flujo Call Gapping. De la misma forma utilizando (13) tenemos que :

Retardo de dos accesos con $t_{db} = 8\text{ms.}$

$$R(2) = (175)(1.25\text{ms.}) + 100\text{mspll} + (3)(3.33\text{mspll.}) + (2)(75)(8\text{ms.}) + 0.25\text{spll} = 1.7787\text{spll.}$$

En donde 1.7787spll. es el retardo medio que sufre una llamada inteligente que requiere de dos accesos a la base de datos utilizando Call Gapping.

Resumiendo:

$$R(0) = 0.57208\text{spll.}$$

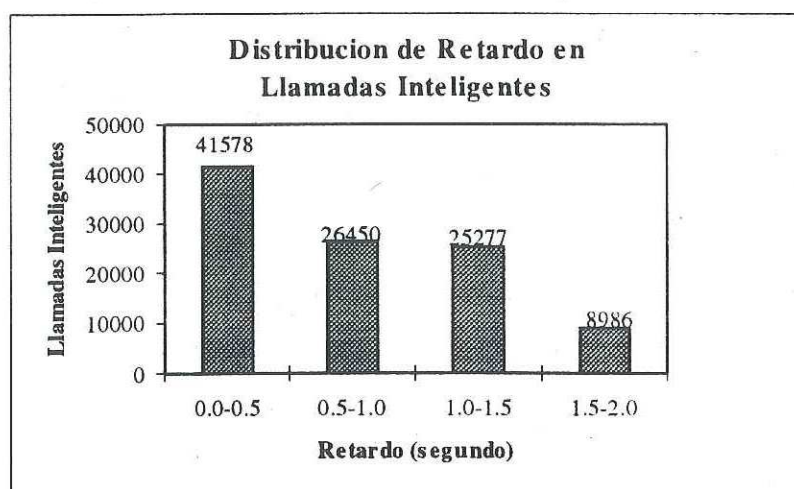
$$R(1) = 1.10041\text{spll.}$$

$$R(2) = 1.62879\text{spll.}$$

En donde $R(0)$, $R(1)$ y $R(2)$ son los retardos que sufre una llamada inteligente al acceder al SCP, y requiriendo cero, uno o dos accesos a la base de datos respectivamente, en período de alto tráfico en donde la capacidad del sistema es alcanzada.

A diferencia de sin control y control de la Ventana, en Call Gapping no podemos identificar claramente de la figura 52 los tres grupos de retardo de llamadas en momentos de alto tráfico, identificando con ello el escenario de llamadas inteligente usado. Sin embargo, podemos estar seguros de que los valores calculados son la media de retardo que sufre una llamada inteligente en momentos de alto tráfico.

La figura 53 muestra un histograma de la distribución de retardo en las llamadas inteligentes. Podemos ver que tenemos tanto llamadas marcadas como bloqueadas, así como llamadas marcadas como pérdidas. Las llamadas bloqueadas surgen debido a los mensajes de control que ajustan la tasa de salidas en el SSP y éste empieza a almacenar llamadas inteligentes en su buffer. Y las llamadas pérdidas son causadas por lento funcionamiento del mecanismo en donde entre intervalos de muestreo, puede estar el buffer del CP congestionado empezando a descartar llamadas inteligentes previamente admitidas por el sistema.



#Llamadas Pérdidas = 395

#Llamadas Bloqueadas = 14693

Figura 53 Distribución de retardo de llamadas Inteligentes con Call Gapping.

Del histograma podemos ver que el 12.5% de las llamadas generadas fueron bloqueadas y que el 0.38% de las llamadas admitidas por el sistema fueron pérdidas, esto significa que el 99.62% fueron atendidas satisfactoriamente cumpliendo con la calidad de servicio requerida por una llamada inteligente. Por lo tanto, el desempeño del mecanismo se considera aceptable en nuestro estudio. Sin embargo, lo óptimo en nuestro estudio es no tener llamadas inteligentes pérdidas una vez que fueron admitidas y enviadas al SCP en espera de ser servidas y que la probabilidad de bloqueo en el SSP disminuya conservando la calidad de servicio inteligente.

V.5.3 GOODPUT EN EL SCP

En la figura 54, podemos observar el Goodput monitoreado en un CP que implementa el mecanismo de control de flujo Call Gapping, como parámetro de desempeño del sistema.

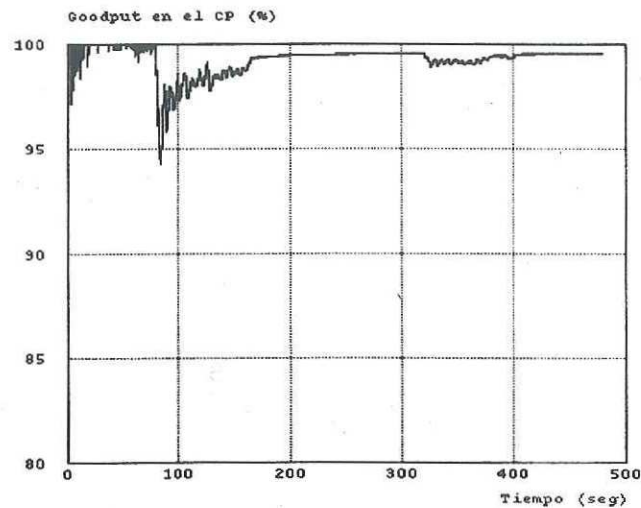


Figura 54 Goodput en el CP con Call Gapping.

De la figura podemos analizar que el Goodput antes de que el tráfico aumentara permanecía en un buen nivel de desempeño aproximadamente a 100%. Sin embargo, en el momento en que el tráfico alcanza la capacidad máxima del sistema a 80 segundos de simulación, el Goodput decrementa hasta por debajo de un 95 %, esto debido a la lenta reacción del mecanismo para ajustar el tráfico enviado desde el SSP, cayendo en posible estado de congestión en tiempos en que el intervalo de muestreo aun no se cumple. Después de que el mecanismo reacciona, éste trata de ajustar debidamente el tráfico, tratando de normalizar el caudal de llamadas a la capacidad del sistema. Sin embargo, debido a que en el primer período de sobreflujo hubo pérdidas por congestión, el Goodput no se normaliza al 100%. De igual manera, en el segundo período de alto tráfico, a 320 segundos de simulación, el Goodput vuelve a caer; pero no tanto como anteriormente, esto debido a que el intervalo de muestreo alcanzó a reaccionar adecuadamente.

V.6 FASE II: MECANISMO DE CONTROL DE CONGESTIÓN A DOS NIVELES.

Recordemos que el objetivo de esta segunda fase, es explorar una base de datos remota compartida por todos los CP de los SCP que conforma la configuración bajo estudio. En donde se hace uso de la topología dual Token Ring de alta velocidad para enlazar a los CPs con la base de datos remota. La base de datos remota es usada para disminuir el tiempo de respuesta sufrido por una llamada inteligente en momentos de alto tráfico, manteniendo un mayor caudal eficaz.

De este modo, se empleará el mecanismo de control de congestión a dos niveles para controlar la utilización de la base de datos remota, compartiendo la carga en momentos de alto tráfico.

Para llevar a cabo la simulación se usa el modelo de tráfico mostrado en la figura 30, el escenario de llamadas de la tabla IX, los parámetros generales mostrados en la tabla III, y los parámetros óptimos encontrados para utilizar el mecanismo de control de congestión a dos niveles mostrados en la tabla VIII.

V.6.1 DETECCIÓN DEL CUELLO DE BOTELLA EN EL SCP.

Las figuras 55, y 56, nos presentan el número de llamadas inteligentes almacenadas en el CP en espera de ser servidas por el CPU y la base de datos (LDB), respectivamente durante el tiempo de simulación. Del mismo modo la figura 57, muestra el total de las llamadas inteligentes almacenadas en el buffer del CP.

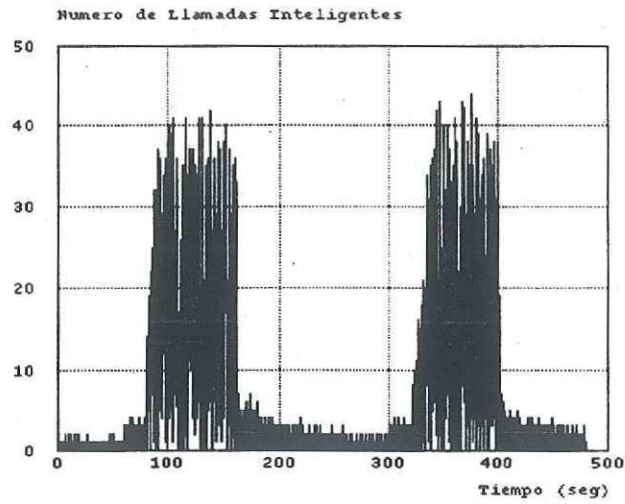


Figura 55 *Número de llamadas en espera de ser atendidas por el CPU con control de congestión a dos niveles.*

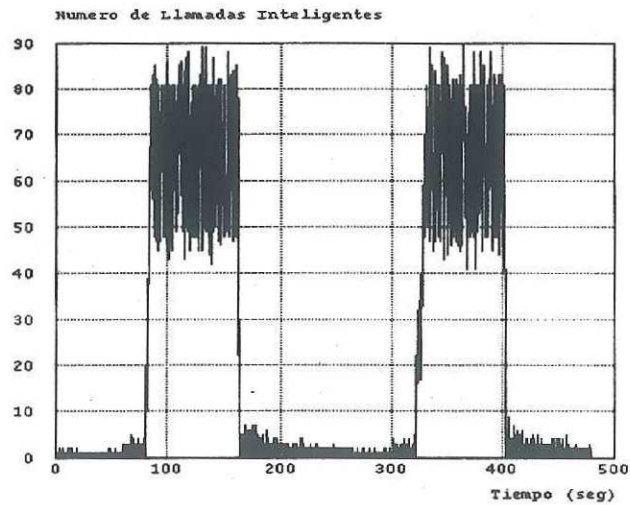


Figura 56 *Número de llamadas en espera de ser atendidas por LDB con control a dos niveles*

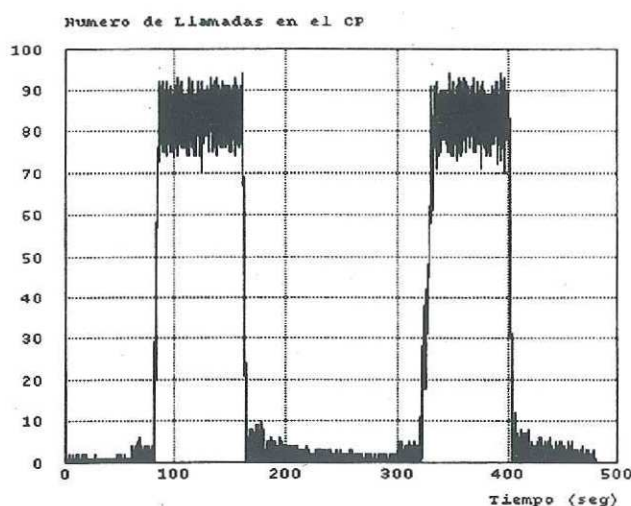


Figura 57 *Número de llamadas en el CP con respecto al tiempo de simulación con control a dos niveles.*

En la figura 56, podemos ver que el número de llamadas almacenadas en espera de ser servidas por la base de datos fluctúa entre 80 y 50 llamadas inteligentes, esto debido al mecanismo de control de sobre flujo implementado en el CP, el cual forma el segundo nivel del mecanismo bajo estudio, con el fin de compartir la carga con la base de datos remota (RDB) en momentos de alto tráfico. De este modo, el CP monitorea el buffer cada vez que una llamada arriba, y si el número de llamadas que requieren de acceder a la base de datos sobrepasa un valor de umbral superior (80 llamadas) el CP empieza a direccionar llamadas inteligentes a ser servidas por la base de datos remota, al mismo tiempo que puede direccionar a la base de datos local, hasta el momento en que el número de llamadas inteligentes en el buffer del CP, que requieren acceder a la base de datos, sea detectado por el umbral inferior (50 llamadas), el CP para de enviar llamadas inteligentes hacia la base de datos remota compartida.

Por otra parte, de la figura 55, podemos ver que el número de llamadas que solo requieren de ser procesadas por el CPU, para que después sean enviadas a la base de datos local, o a la base de datos remota, o como respuesta hacia el SSP, se ve incrementado casi a una media de 30% de la capacidad máxima del buffer en el CP. La figura 57, muestra el total de llamadas inteligentes almacenadas en el CP, en donde nos damos cuenta de que la media de llamadas no sobrepasa un 90% de la capacidad máxima del buffer del CP, esto debido al mecanismo de control de flujo de la Ventana utilizado en el primer nivel, el cual ya se estudió en forma separada.

Por otra parte, las figuras 58, 59, y 60, nos presentan la utilización con respecto al tiempo de simulación del CPU, la base de datos local y de la base de datos remota, respectivamente.

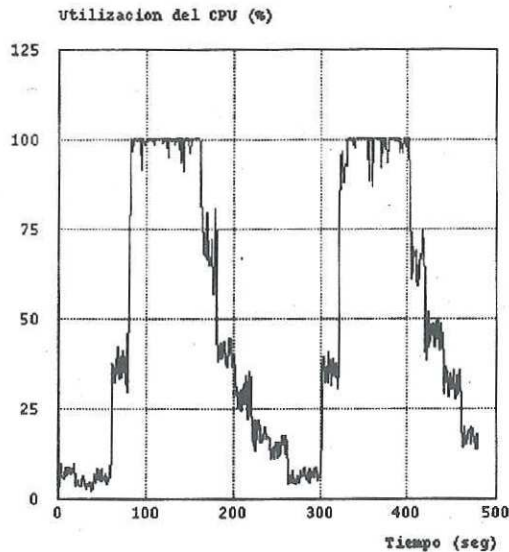


Figura 58 Utilización del CPU

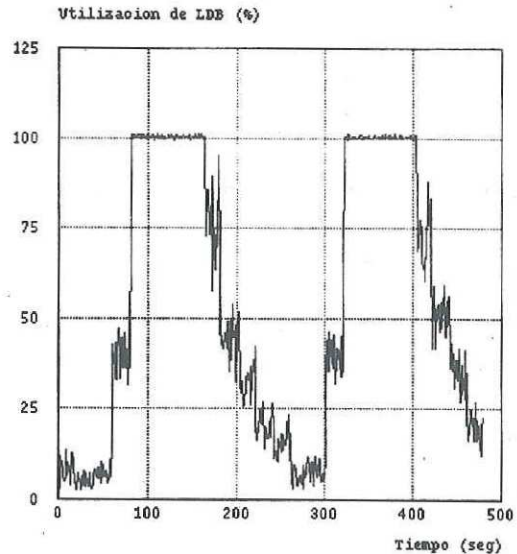


Figura 59 Utilización de LDB.

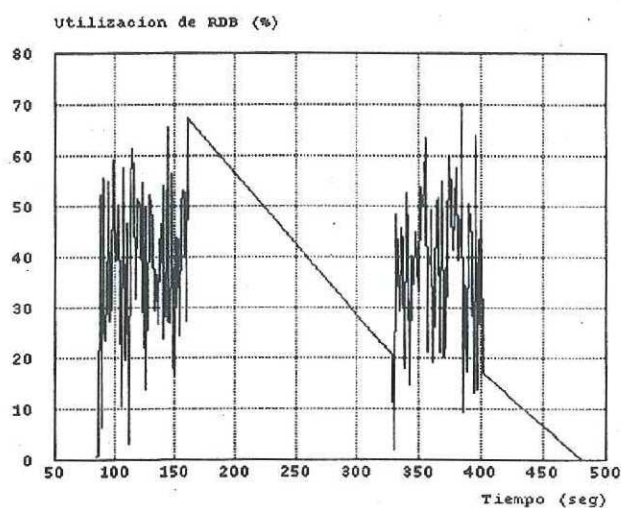


Figura 60 Utilización de RDB con el mecanismo de control de congestión a dos niveles.

De las figuras podemos ver que tanto el CPU, como la base de datos local están al 100% en período de alto tráfico. Por lo tanto, la utilización del CPU se incrementa debido al uso de la base de datos remota para el procesamiento de llamadas inteligentes, en donde toda llamada después de ser atendida por la base de datos se regresa a ser de nuevo procesada por el CPU, entonces como el caudal en el SCP se elevó, un número mayor de llamadas son procesadas y gracias al mecanismo de control de congestión a dos niveles todas las llamadas son servidas satisfactoriamente. Por otra parte, hemos detectado que el cuello de botella continua en la base de datos local debido a su lento procesamiento provocando que una media de 70% de llamadas se almacenen en el buffer del CP. Sin embargo, la utilización de la base de datos remota como lo muestra la figura solo se incrementa a una media del 40% en los periodos de alto tráfico, se podría incrementar mas pero esta cantidad es la adecuada, porque de lo contrario, tendríamos el problema de cuello de botella en dos entidades en el sistema, en el CPU por el gran número de llamadas

procesadas y en la base de datos local por su lento procesamiento de servicio.

V.6.2 RETARDO DE LLAMADAS INTELIGENTES

La figura 61, muestra el retardo por llamada inteligente de una manera dinámica con respecto al tiempo de simulación. En donde podemos ver que en momentos de alto flujo de tráfico las llamadas inteligentes se ven incrementadas en su tiempo de respuesta. Debido a que este mecanismo hace uso del mecanismo de control de flujo de la Ventana en el primer nivel, el cual como ya nos dimos cuenta, nos garantiza que toda llamada enviada al SCP será atendida satisfactoriamente, por lo tanto esperamos entonces que aquellas llamadas que no requieren de acceder a la base de datos en período de alto tráfico se vean afectadas por un retardo extra, esto debido al funcionamiento del mecanismo utilizado como primer nivel.

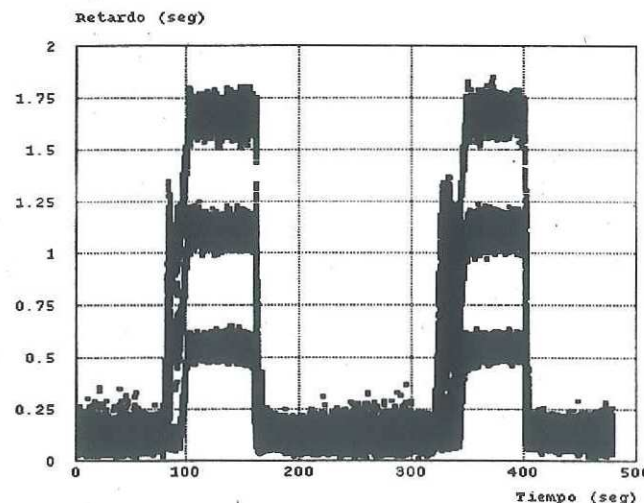


Figura 61 Retardo de Llamadas Inteligentes con control de congestión a dos niveles.

El retardo de las llamadas antes de que la cantidad de tráfico generado no llegue a la capacidad del sistema, sufren el mismo tiempo de retardo estudiado anteriormente en donde

$R(0)$, $R(1)$ y $R(2)$ son los retardos de una llamada inteligente que requiere de cero, uno o dos accesos a la base de datos, respectivamente. Entonces tenemos:

$$R(0) = 0.10458 \text{ spll.}$$

$$R(1) = 0.11491 \text{ spll.}$$

$$R(2) = 0.12524 \text{ spll.}$$

Por otra parte, cuando el mecanismo del primer nivel detecta un posible estado de congestión o cuando monitorea que tiene mas de 100 llamadas fuera en espera de ser atendidas, el SSP cierra su ventana de transmisión. En este tiempo, el SCP tiene un número de llamadas almacenadas en donde éstas requieren de acceder a la base de datos, provocando con esto un incremento en el tiempo de retardo. Sin embargo, el tiempo de retardo es menor al experimentado con control de la ventana y Call Gapping, debido a la activación del mecanismo de control del segundo nivel en donde cada CP del SCP empieza a enviar llamadas inteligentes a la base de datos remota compartida para que sean servidas y después devueltas para procesarse de nuevo por el CPU, en donde éste verifica si la llamada requiere de más información para llevar a cabo el servicio inteligente, en donde si es así, se envía de nuevo a ser procesada por la base de datos (local o remota). De otro modo, si la llamada ya no requiere de mas información, es enviada al SSP para que continúe con el servicio inteligente.

En momentos en donde las bases de datos local y remota son utilizadas para el procesamiento de una llamada inteligente, el buffer del CP que contiene a la mayoría de las llamadas inteligentes que requieren de acceder la base de datos, se encuentra en una media de 82 llamadas, de tal manera que el retardo extra que sufre una llamada inteligente que

espera en el SSP a ser enviada al SCP, se determina utilizando la ecuación (10) como:

$$R_e = B_{cp} * t_{cpu} = (82) (3.33ms) = 0.27306 \text{ spll.}$$

Así, 0.27306spll. es el tiempo mínimo que debe de esperar una llamada inteligente para ser procesada por última vez por el CPU y mandada como respuesta hacia el SSP en donde al recibir una respuesta abre temporalmente su ventana y envía una nueva llamada inteligente hacia el SCP.

De este modo, utilizando (11), (12) y (13), podemos calcular el tiempo medio de retardo por llamada inteligente que requiere de cero, uno o dos accesos a la base de datos en momentos de alto tráfico, considerando que debido al uso del mecanismo de control de congestión a dos niveles se monitorea una longitud media del buffer en el SSP de 150 llamadas ($B_{ssp} = 150$ llamadas) en momentos de alto tráfico.

Utilizando (11) tenemos el retardo de llamadas con cero acceso en momentos en que el tráfico sobrepasa la capacidad del sistema:

$$R(0) = (150)(1.25ms.) + 100mspll + 3.33mspll. + 0.27306spll. = 0.56389spll.$$

De la misma manera se calcula el retardo para llamadas que requieren de un acceso a la base de datos para poder a completar el servicio con (12) :

Retardo de un acceso con $t_{db} = 8ms$.

$$R(1) = (150)(1.25ms) + 100mspll + (2)(3.33mspll) + (82)(8ms) + 0.27306spll = 1.2232spll.$$

En donde 1.2232spll. representa el tiempo que tarda una llamada inteligente en ser servida por el SCP cuando requiere de un acceso a la base de datos. De la misma forma utilizando (13) tenemos que para:

Retardo de dos accesos con $t_{ldb} = 8\text{ms}$.

$$R(2) = (150)(1.25\text{ms}) + 100\text{mspll} + (3)(3.33\text{mspll}) + (2)(82)(8\text{ms}) + 0.2730\text{spll} = 1.8825\text{spll}.$$

En donde 1.8825spll. es el retardo que sufre una llamada inteligente que requiere de dos accesos a la base de datos.

Resumiendo:

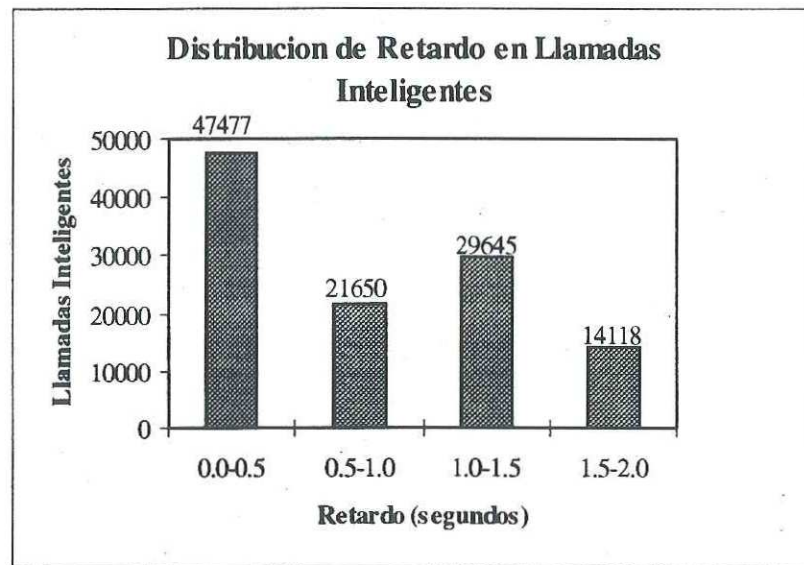
$$R(0) = 0.56389\text{spll}.$$

$$R(1) = 1.22322\text{spll}.$$

$$R(2) = 1.8825\text{spll}.$$

En donde $R(0)$, $R(1)$ y $R(2)$ son los retardos que sufre una llamada inteligente que requiere de cero, uno o dos accesos a la base de datos, bajo el mecanismo de control de congestión a dos niveles. De otro modo $R(0)$, $R(1)$ y $R(2)$ pueden ser aproximadamente identificándose la figura 61, en momentos de alto tráfico, esto debido al escenario de llamadas inteligente utilizado.

En la figura 62, se muestra un histograma de la distribución de retardo de llamadas inteligentes utilizando el mecanismo bajo estudio. En donde podemos apreciar que el caudal de llamadas incremento y el retardo disminuyo relativamente.



#Llamadas Pérdidas = 0 #Llamadas Bloqueadas = 4296

Figura 62 Distribución de retardos de llamadas inteligentes con control de congestión a dos niveles.

De la figura podemos ver que un 3.66% de llamadas generadas fueron marcadas como bloqueadas, esto representa que el sistema fue capaz de procesar mas de 50% de las llamadas bloqueadas a comparación de los otras configuraciones, además que el 100% de las llamadas admitidas y enviadas hacia el SCP en espera de ser atendidas fueron servidas satisfactoriamente. Por lo tanto, este novedoso mecanismo de control utilizado, junto con la propuesta de la utilización de la base de datos remota compartida, son los óptimos considerados en nuestra investigación cumpliendo con todos los objetivos propuestos.

V.6.3 GOODPUT EN EL SCP

En la figura 63, se muestra el Goodput monitoreado en un CP que conforma a un SCP que implementa el mecanismo de control de congestión a dos niveles.

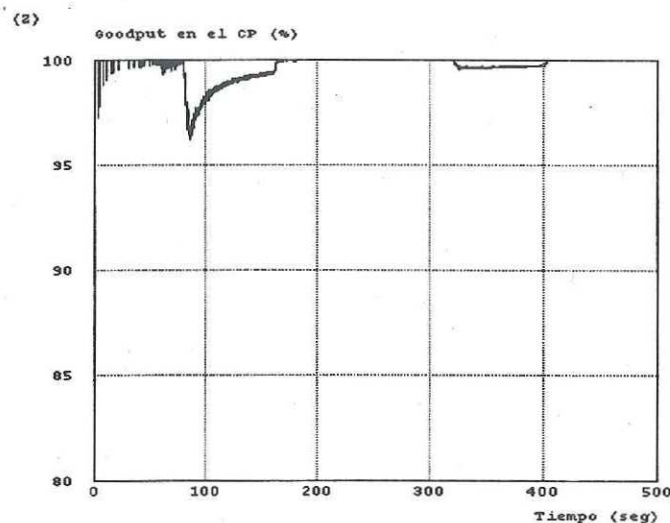


Figura 63 Goodput en el CP con mecanismo de control de congestión a dos niveles.

De la figura podemos ver que a 80 segundos de tiempo de simulación, el Goodput decrementa por el almacenamiento de llamadas inteligentes, pero rápidamente se normaliza, manteniéndose en un 100% en el resto del tiempo de simulación, representando con esto que el caudal eficaz en el sistema aumento y que todas las llamadas fueron servidas satisfactoriamente.

V.7 DIFERENCIAS ENTRE WINDOW, CALL GAPPING Y DOS NIVELES.

Sin duda, después de analizar detenidamente los resultados obtenidos de simular cada uno de los mecanismo de control sobre impuestos en nuestra configuración de red inteligente, hemos encontrado que la implementación del uso de una base de datos remota

(compartida) es una muy buena opción. Sin embargo, este esquema no es funcional sin el uso en conjunto del mecanismo de control de congestión a dos niveles propuesto en este trabajo de tesis como una alternativa futura cuando los servicios de red inteligente requieran de una elevada complejidad y un gran volumen de información para llevarse a cabo.

Sin embargo, a continuación describiremos algunos puntos claves en la diferencia entre los mecanismos de control de flujo estudiados anteriormente.

Control de la Ventana

- Este mecanismo no requiere ninguna funcionalidad llevada a cabo por el SCP y este mecanismo es implementado en el SSP.
- Este mecanismo esta basado en envío-respuesta, por lo tanto apriori podemos decir que es un mejor control en la carga impuesta al SCP, sin embargo, el SSP requiere de la utilización de temporizadores los cuales son activados por cada llamada enviada al SCP.
- Este mecanismo garantiza que todas las llamadas admitidas a ser servidas por el SCP serán servidas exitosamente dentro de límite de tiempo de respuesta. Siempre y cuando el sistema este libre de fallas.

Control de Call Gapping

- El SCP debe de llevar a cabo estadísticas detalladas y medir el flujo de llamadas entrantes enviadas por cada SSP periódicamente, y generar un mensaje de control hacia el SSP si el flujo de llamadas sobrepasa un umbral. Por lo tanto, este mecanismo es implementado en el SCP.

- En caso de cambios abruptos de flujo de tráfico, Call Gapping puede no ser capaz de adaptarse. Debido a que éste es activado dependiendo de las estadísticas tomadas por el SCP sobre condiciones de carga al inicio de un intervalo de tiempo.
- Debido al largo tiempo de reacción de este mecanismo, pueden perderse llamadas en el SCP. Como también debido a los límites sobre impuestos al SSP pueden haber llamadas bloqueadas. Por lo tanto, esto no garantiza que todas las llamadas aceptadas por el sistema serán exitosamente servidas respetando el tiempo de respuesta inteligente.

Mecanismo de Control de Congestión a dos niveles: Ventana y Sobre Flujo

- Este mecanismo hace uso del control de la Ventana en el primer nivel.
- Este mecanismo es implementado tanto en el SSP como en el SCP.
- El SCP debe de monitorear la longitud de su buffer continuamente. Y activar el mecanismo de sobre flujo si el número de llamadas sobre pasa un valor de umbral.
- Este mecanismo garantiza que todas las llamadas admitidas en el sistema sean servidas exitosamente.
- Bajo el uso de este mecanismo el sistema es capaz de proveer servicios a una gran cantidad de llamadas inteligentes en periodos de alto flujo de tráfico, cumpliendo con la Calidad de Servicio requerida.

Aparte de las diferencias anteriormente mencionadas, por medio de las siguientes figuras podemos remarcar lo siguiente:

La figura 64, muestra una ampliación del estado del buffer en un período de alto tráfico en donde el sistema se encuentra cargado a su máxima capacidad, utilizando la configuración de red inteligente sin control, con Ventana (Window), con Call Gapping y con el mecanismo a dos niveles, respectivamente. Así, de la figura podemos ver que: sin control mantiene continuamente al CP en problemas de congestión, descartando llamadas inteligentes y con esto decrementando el caudal en el sistema representado por un decremento en el Goodput. Control de Ventana mantiene el buffer del CP a una media constante fuera de problemas de congestión y de este modo manteniendo un Goodput adecuado. El mecanismo a dos niveles por hacer uso del control de la ventana en el primer nivel, mantiene al buffer en un estado similar al mecanismo de la Ventana, pero con un caudal mas elevado, pudiendo servir mayor número de llamadas y tener mas de un 50% menos de llamadas bloqueadas en el SSP, debido al uso de la base de datos remota. Por último Call Gapping, debido a la lenta reacción de su control mantiene al buffer variando en momentos de congestión y en momentos de bajo caudal, provocando con esto, que Call Gapping no garantice que todas las llamadas enviadas al SCP se cumplan.

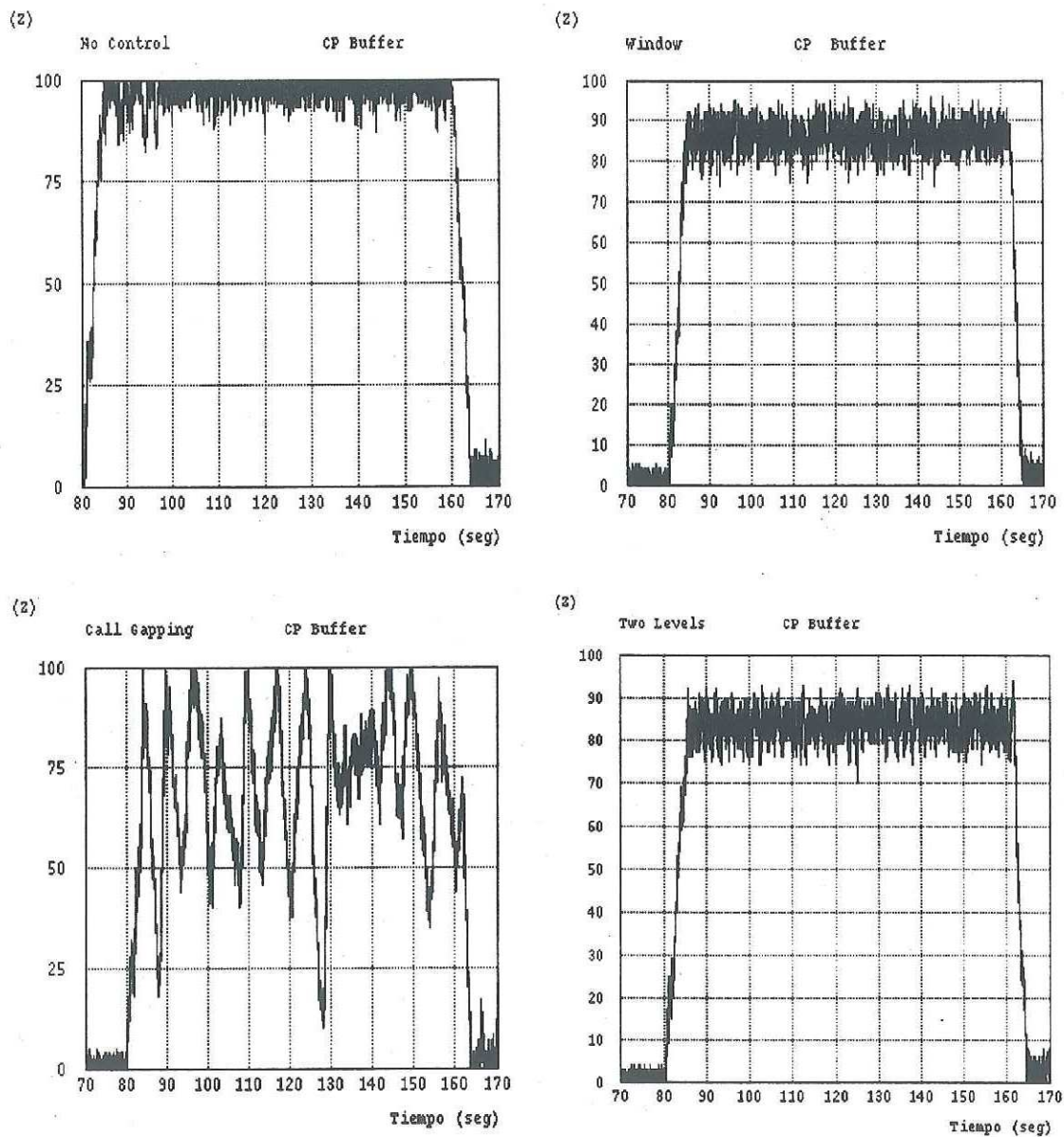


Figura 64 Comparación del estado del buffer en el CP utilizando los mecanismos de control bajo estudio y la configuración de red inteligente sin control.

La figura. 65 muestra la comparación de Goodput obtenido en cada uno de los mecanismos comparados con “sin control” en la configuración de red inteligente utilizada en nuestra investigación.

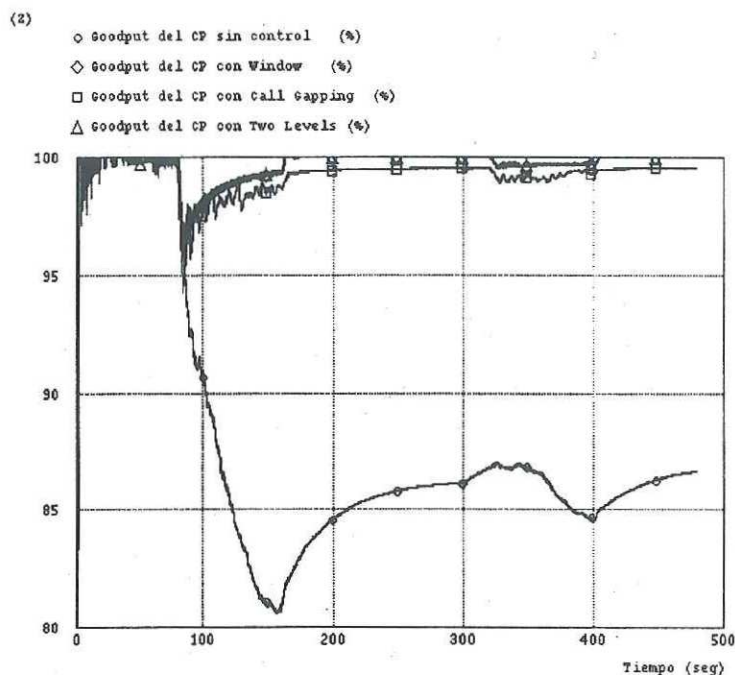


Figura 65 Comparación del Goodput en el CP sin control contra los mecanismos de control estudiados.

De la figura podemos ver, en el caso del sistema “sin control”, el Goodput decrementa considerablemente representando un estado de cuello de botella en el sistema provocando un almacenamiento masivo en el buffer llegando a un posible estado de congestión, descartando nuevas llegadas de llamadas. Sin embargo, claramente se ven en la figura los beneficios en el uso de un mecanismo de control, los cuales mantienen el Goodput a un nivel considerablemente bueno en donde el problema de cuello de botella está controlado y con ésto se minimiza la probabilidad de bloqueo en el SCP, permitiendo cumplir con la calidad de servicio requerida por la red inteligente.

VI. CONCLUSIONES, RECOMENDACIONES Y COMENTARIOS

VI.1 CONCLUSIONES

Este trabajo de tesis ha sido enfocado al estudio de mecanismos de control de flujo para proporcionar de manera eficaz y robusta servicios en la Red Inteligente. En un futuro muy cercano la introducción de un gran número de novedosos y complejos servicios en la Red Inteligente implicarán no solo un incremento en el tráfico de señalización, sino también un incremento en la carga ofrecida hacia el nodo central SCP, provocando con ésto problemas de cuellos de botellas que ocasionan un retardo adicional inaceptable en el tiempo de establecimiento de una llamada inteligente, reduciendo así el caudal eficaz en la red.

Para solventar los problemas derivados del cuello de botella, incrementar el caudal eficaz de llamadas inteligentes y al mismo tiempo cumplir con los requerimientos de calidad de servicio en la Red Inteligente, en este trabajo de tesis se estudiaron tres mecanismos de control comparándose con la situación en que no hay ningún control implementado. Estudios anteriores no detallaban el modelo de multiprocesadores en el nodo centralizado SCP, ni el uso de una base de datos remota en momentos de sobre flujo. En nuestro estudio hemos utilizado una arquitectura de Red Inteligente (Figura 20) en la cual el SCP esta compuesto por múltiples procesadores de llamadas participando en el procesamiento de llamadas inteligentes. Cada procesador de llamada está compuesto por una base de datos local la cual es accesada varias veces dependiendo de la complejidad del servicio inteligente. Los procesadores de llamadas están enlazados por una red dual Token Ring de alta velocidad compartiendo el acceso a una base de datos remota. Para llevar acabo el estudio

de esta arquitectura multiprocesador de RI se utilizó el simulador orientado a comunicaciones OPNET, en donde se implementaron modelos de colas representando tres esquemas de control impuestos al SCP, llamados, Control de Banda de Llamadas (Call Gapping), Control de la Ventana (Window) y Control de Congestión a dos niveles. Con el fin de evaluar el desempeño del sistema se crearon varios modelos de redes de colas como el que se presenta en la figura 21. Hemos estudiado el comportamiento del sistema utilizando un escenario de llamadas inteligentes el cual engloba la complejidad de los servicios inteligentes a un plazo futuro y considerando factores tales como:

- El tiempo de respuesta a una petición de servicio inteligente,
- el comportamiento del buffer en el CP, comparando el número de llamadas que se encuentran en espera de ser atendidas por el CPU o por la base de datos local,
- el porcentaje de utilización del CPU y la base de datos local, para identificar con esto la entidad que nos proporciona el problema del cuello de botella,
- el número de llamadas pérdidas o bloqueadas en el sistema, y
- el Goodput en el CP, representando el nivel de cuello de botella y a la vez el caudal eficaz en el SCP.

Con lo anterior hemos evaluado cada uno de los mecanismos de control comparándolos con la situación en la que no hay ningún control implementado, teniendo como resultados los encontrados en la tabla X.

Tabla X *Tabla comparativa de los mecanismos de control y sin control.*

Arquitectura	Probabilidad de Bloqueo	# Llamadas con tiempo de respuesta ≤ 2 seg.	Cumple con la Calidad de Servicio requerida por la RI.
Sin Control	13%	102195	No
Ventana	10%	105972	No
Bandas de Llamadas	12.5%	102291	No
Dos Niveles	3.66%	112890	<<Si>>

En la tabla X podemos ver como el mecanismo de control de congestión a dos niveles cumple con los requerimientos de calidad de servicio de la RI, en donde se indica que el 95% de las llamadas deben de ser servidas satisfactoriamente en un tiempo no mayor de 2 segundos, el 5% restante representan la probabilidad de bloqueo o llamadas que fueron servidas en un tiempo mayor a 2 segundos.

VI.2 RECOMENDACIONES

Se propone las siguientes recomendaciones para trabajo futuro, las cuales pueden ser implementadas y estudiadas utilizando los mismos modelos de red creados en nuestro estudio, como son:

- La implementación de diferentes mecanismos que manejen el tamaño de la ventana. Como por ejemplo, un mecanismo similar a Slow Start implementado en TCP.
- Uso de tráfico de fondo en la red dual Token Ring utilizado para enlazar a los CPs del SCP y a la base de datos remota. Este tipo de tráfico se puede identificar con funciones

de administración en la Red Inteligente.

- Modelar el uso de varios SSPs compartiendo un solo nodo SCP.
- Modelar la probabilidad de fallas en la base de datos local en un CP con el fin de enrutar las llamadas hacía otros CPs, a través de la red dual Token Ring.
- Modelar la combinación de llamadas inteligentes y no inteligentes haciendo uso del SSP, llevando a cabo la interconexión entre el abonado que llama y el abonado a quien se le llama al mismo tiempo que se utiliza el mecanismo de control propuesto.
- Estudiar el efecto de interoperabilidad entre diferentes entidades físicas que conforman la Red Inteligente a parte del SSP y SCP con el fin de llevar a cabo el modelado de servicios inteligentes mas complejos. Por ejemplo aquellos servicios que hacen uso de periféricos inteligentes o adjuntos.

VI.3 COMENTARIOS

A lo largo de la elaboración de este trabajo de tesis, nos dimos cuenta de que la Red Inteligente es una arquitectura de moda en las redes de telecomunicaciones, la cual se trata de expandir para usos futuros como una dorsal de telecomunicaciones. La Red Inteligente es en la actualidad una área de investigación activa con problemas abiertos. Siendo el estudio de mecanismos de control de flujo tan solo una parte de éstos.

Por lo tanto, unas de las principales áreas de interés para continuar con la investigación en el área y las cuales pueden servir como temas de tesis son:

- La interoperabilidad de la Red Inteligente con otras redes.
- La Red Inteligente de Banda Ancha, manteniendo un sin número de servicios

transportando tanto voz, datos, vídeo e imagen.

- Los sistemas de creación de servicios inteligentes utilizados en la Red Inteligente.

Siendo este un tema de mucho interés en compañías que proporcionan servicios de red inteligente.

LITERATURA CITADA

Audestad Jan A., "Intelligent Networks Overview", <http://www.nta.no/teletronikk/>, 1993.

Baver Helen A., J. Kulzer., E. Sable, "Designing Service-Independent Capabilities for IN",
IEEE Communications Magazine, pp. 31-41, December 1992.

Berger Arthur W., "Comparison of Call Gapping and Percent Blocking for Overload
Control in Distributed Switching Systems and Telecommunications Networks", IEEE
Transactions on Communications, Vol. 39, No. 4, pp. 574-580, April 1991.

Berman Roger K., J. Brewster, "Perspective on the AIN Architecture", IEEE
Communications Magazine, February 1992.

Chung M.Y., D. K. Sung, "Performance Evaluation of Intelligent Networks
Accommodating Varios IN Services as well as Basic ISDN Services", Proceedings of
ICCC'95, Vol. 1, 12 va. Conferences, pp. 195-200, Seul Korea, 1995.

Duran José M., J. Visser, "International Standars for IN", IEEE Communications Magazine,
pp. 34-42, February 1992.

Fukuda Akira, "Input Regulation Control Based on Periodical monitoring Using Call Gapping Control", Script Technical, Inc. 1986, Electronics and Communications in Japan, Part 1, Vol. 69., No. 11, pp. 84-93, 1986.

Jabbari Bijan, "Common Channel Signaling System Number 7 for ISDN and Intelligent Networks", Proceedings of the IEEE, Vol. 79, No. 2, pp. 155-169, February 1991.

Jain Raj, "A Timeout-Based Congestion Control Scheme for Window Flow - Controlled Networks", IEEE Journal on Selected areas in communications, Vol. SAC-4, No. 7, October 1986.

Modaressi R., A. Skoog, "Signaling System No. 7 : A tutorial", IEEE Communications Magazine, pp. 19-35, July 1990.

OPNET, manuales de usuario o manual en línea. Versión 2.5B, 1995.

Ouellette M., L. O. Barbosa, "Performance Evaluation of a Multiprocessor SCP for IN services", Proceedings of IEEE CCECE'97, Vol. II, pp. 804-807, May 1997.

Peterson Larry, D. Bruce, "Computer Networks : A System Approach", Morgan Kaufmann Publisher, Inc. 1996.

Pham Huy X., R. Betts, "Congestion Control for Intelligent Networks", Computer Networks and ISDN System, pp. 511-524, January 1994.

Pouzin Louis, "Methods, Tools, and Observations on Flow Control in Packet Switched Data Networks", IEEE Transactions on Communications, Vol. com-29, No. 4, April 1981.

Q.1201 ITU-T Recommendation, "Principles of IN Architecture", October 1992.

Q.1202 ITU-T Recom., "IN Service Plane Architecture", October 1993.

Q.1203 ITU-T Recom., "IN Global Functional Plane Architecture", October 1994.

Q.1204 ITU-T Recom., "IN Distributed Functional Plane Architecture", April 1993.

Q.1205 ITU-T Recom., "IN Physical Plane Architecture", March 1996.

Q.1208 ITU-T Recom., "INAP", March 1993.

Q.700 ITU-T Recom., "Introduction to CCITT SS7 FASCICLE", November 1988.

Robart Lewis, "Is There A Future for Global Intelligent netowrks Standars?", Proceedings of Workshop IN'97, Colorado, Spring, May 1997.

Robrock Richard B., "The Intelligent Network- Changing the face of Telecommunications", Proceedings of the IEEE, Vol. 79, No. 1, January 1991.

Smith E. Donald, "Ensuring Robust Call Throughput and Fairness for SCP Overload Controls", IEEE/ACM Transactions on Networking, Vol. 3, No. 5, October 1995.

Tanenbaum, "Computer networks", Prentice Hall, Third Edition, 1997.

Thorner Jan, " Intelligent Network", Artech House, Inc. Primera edicion, Boston, 1994.

Unger Stefan, "Standardization of INAP The European Approach", Proceedings of Workshop IN'97, Colorado Spring, May 1997.

ANEXO A

DESCRIPCIÓN GENERAL DEL MÉTODO DE SIMULACIÓN OPNET

OPNET (OPTIMIZED NETWORK ENGINEERING TOOLS)

OPNET es un sistema de ingeniería de tráfico capaz de simular grandes redes de comunicaciones con modelos detallados de protocolos y análisis de desempeño (OPNET, 1995). Las características principales de OPNET son:

Orientado a objetos: Los modelos en OPNET son especificados en términos de objetos, cada uno con un conjunto de atributos configurables. El simulador soporta la definición de nuevos objetos con características programables ofreciendo flexibilidad al diseñador.

Especializado en redes de comunicaciones y sistemas de información: Permite la construcción de bloques de modelos enfocados a comunicaciones y procesamiento de información para acelerar el desarrollo de nuevas redes y sistemas distribuidos.

Especificación Gráfica: los modelos son creados vía un editor gráfico el cual provee de herramientas necesarias para especificar el sistema.

Flexibilidad en el desarrollo detallado de modelos: OPNET provee un lenguaje de alto nivel llamado Proto-C con un gran soporte en comunicaciones y sistemas distribuidos. Este ambiente permite el rápido modelado de protocolos de comunicación y algoritmos.

Generación automática de simulaciones: Los modelos creados son automáticamente compilados en archivos ejecutables.

Utilerías integradas para análisis: Evaluación del desempeño, comportamiento de los enlaces, y una gran cantidad de utilerías para estudiar el resultado de la simulación son

proporcionados por OPNET en un ambiente gráfico.

Técnicas avanzadas en identificación de errores: Todas las simulaciones en OPNET automáticamente incorporan utilerías sofisticadas para identificación y búsqueda de errores.

Jerarquía de modelos: Los modelos en OPNET son creados en forma jerárquica para representar redes de comunicaciones. Existen tres niveles en la jerarquía del model. El nivel inferior, conocido como nivel de proceso en donde se codifica el comportamiento de un algoritmo o protocolo en diagramas de transición de estados los cuales son programados con un lenguaje de alto nivel basado en lenguaje C. El nivel medio conocido como nivel de nodo, funciones discretas tales como colas, procesos, transmisores y receptores son definidos en objetos separados. Estos objetos, llamados módulos son creados y modificados usando el editor de nodos en OPNET y éstos combinados entre si forman el mas alto nivel, el nivel de red. La figura 66, nos muestra el forma gráfica la jerarquía de modelos en OPNET.

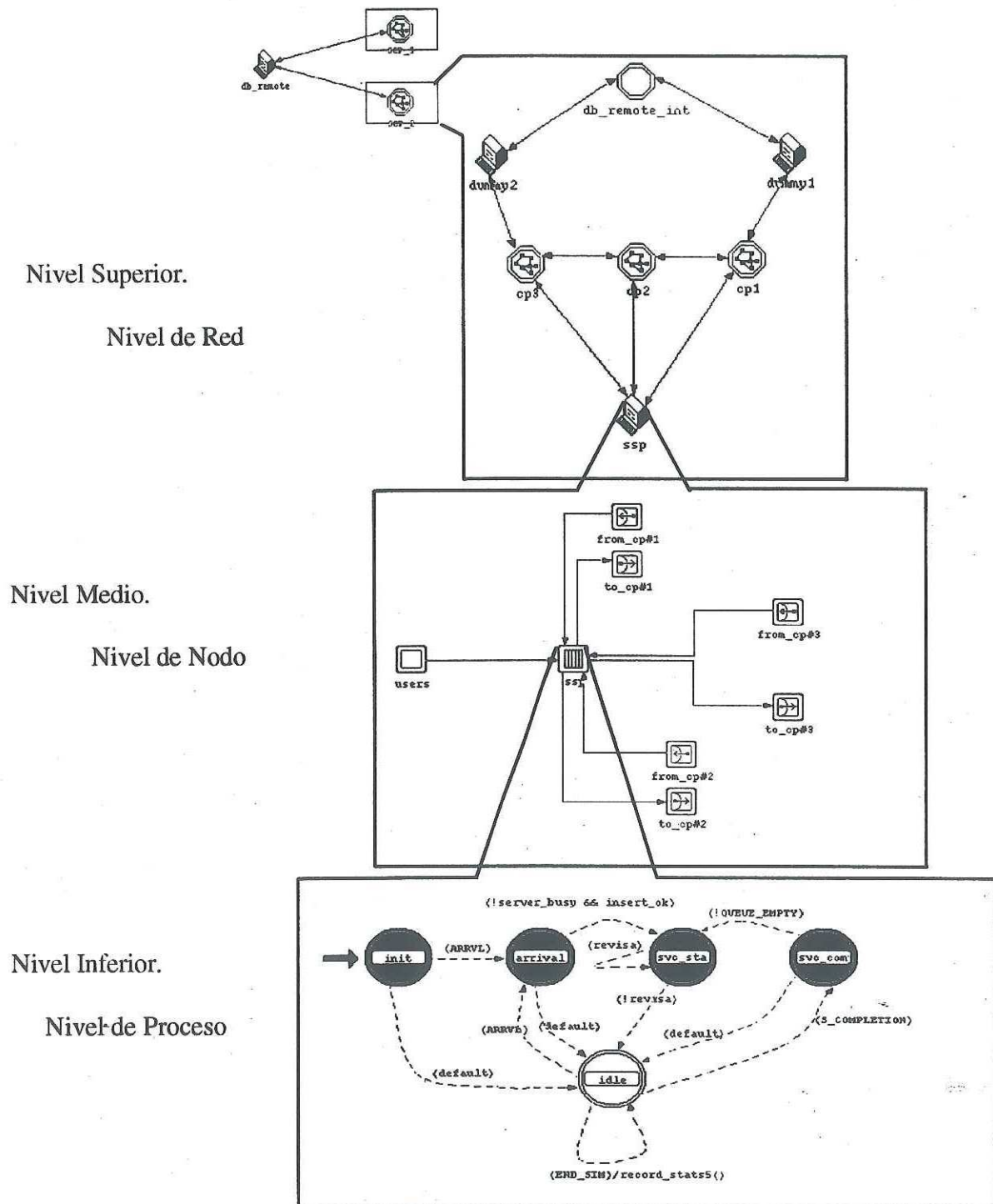


Figura 66 Jerarquía de Modelos en OPNET

GLOSARIO

ABS :	Alternate Billing Service : Servicios de facturación alternativa
AD :	Adjunt : Adjunto
AIN :	Advanced Intelligent Network
ARPAnet :	Advanced Research Projects Agency Network
ASE :	Application Service Element : Elemento de servicio de aplicación
BCP :	Basic Call Procedure : Proceso básico de llamada
CASE :	Computer Aided Software Engineering : Software auxiliar de ingeniería
CCAF :	Call Control Agent Function : Agente de función de control de llamada
CCC :	Credit Card Calling : Llamada con tarjeta de crédito
CCF :	Call Control Function : Función de control de llamada
CCIS :	Common Channel Interoffice Signaling Network : Red de señalización interoficina por canal común
CCSN :	Common Channel Signaling Network : Red de señalización por canal común
CMIP :	Common Management Information Protocol : Protocolo de manejo común de información
CON :	Conference Call, Comunicación tripartita
CP :	Call Processor : Procesador de llamadas
CPH :	Call Party Handling : Manejo de llamada de usuario
CPU :	Central Processor Unit : Unidad de procesamiento central

CRD :	Call Rerouting Distribution : Enrutamiento distribuido de llamadas
CS :	Capabilities Set : Conjunto de capacidades
DCR :	Destination Call Routing : Enrutamiento de llamada a un destino
DDL :	Data Definition Lenguaje : Lenguaje de definición de datos
DFP :	Distributed Functional Plane : Plano Funcional Distribuido
DPC :	Destination Point Code : Código del punto destino
DTMF :	Dual Tone Multi-Frecuency : Marcación de Tonos Multi-Frecuencias
EFSA :	Extended Finite State Automata : Autómata de estados finitos extendida
ETSI :	European Telecommunication Standards Institute : Instituto Europeo de Estandares en Telecomunicaciones
FC :	Functional Component : Componente funcional
FE :	Functional Entity : Entidad funcional
FEA :	Functional Entity Action : Acciones de entidad funcional
FMD :	Follow-Me-Diversion : Siguieme diversión
FPH :	FreePhone : Telefono libre de cobro
GFP :	Global Functional Plane : Plano Global Funcional
GSL :	Global Service Logic : Lógica de servicio global
GTT :	Global Translate Title : Traducción de títulos globales
GUI :	Graphical User Interface : Unidad de interfaz gráfica
HDLC :	High-level Data Link Control : Control de enlace de datos de alto nivel
IN :	Intelligent Network, Red inteligente

INAP :	IN Application Protocol : Protocolo de Aplicación de Red Inteligente
INCM :	IN Conceptual Model : Modelo Conceptual de Red Inteligente
IP :	Intelligent Peripheral : Periférico inteligente
ISDN :	Integrated Service Digital Network : Red digital de servicios integrados RDSI
ISUP :	ISDN User Part : Parte de Usuario RDSI
ITU-T :	International Telecommunication Union - Telecommunication Sector : Unión Internacional de Telecomunicaciones - Sector de Telecomunicaciones
LDB :	Local DataBase : Base de datos local
MAS :	Mass Calling : Llamadas masivas
MCI :	Malicious Call Identification : Identificación de llamadas maliciosas
MTP :	Message Transfer Part : Parte de la transferencia de mensajes
MVI :	Multi-Vendors Interactions : Interacción múlti-vendedores
NAP :	Network Access Point : Punto de acceso a la red
NCP :	Network Control Point : Punto de control de red
NIU :	Network Interface Unit : Unidad de interfaz de red
OA :	Operation Application : Aplicación de operaciones
OPC :	Originating Point Code : Código del punto origen
OSS :	Opertation Support System : Sistema de soporte de operaciones
PC :	Point Code : Codigo Apuntador
PDUs :	Protocol Data Units : Protocolo de unidades de datos
PE :	Physical Entity : Entidad física

PIN :	Personal Identification Number : Número de identificación personal
PSCL :	Portable Service Call List : Lista de llamadas rápidas portables
PSTN :	Public Switched Telephony Network : Red Pública Telefónica Conmutada
QoS :	Quality of Service : Calidad de servicio
RDB :	Remote DataBase : Base de datos remota
RI :	Red Inteligente
RIA :	Red Inteligente Avanzada
SCE :	Service Creation Environment : Punto de creación de servicios
SCEF :	Service Creation Environment Function : Función de creación de servicios
SCEP :	Service Creation Environment Point : Punto de creación de servicios
SCF :	Service Control Function : Función de control de servicios
SCP :	Service Control Point : Punto de control de servicios
SDF :	Service Data Function : Función de información de servicios
SDL :	Specified Description Language : Especificación y descripción de lenguaje
SDP :	Service Data Point : Punto de información de servicios
SF :	Service Feature : Características de servicio
SIB :	Service Independent Block : Bloques independientes de servicios
SLP :	Service Logic Program : Programas lógicos de servicios
SMAF :	Service Management Agent Function : Agente de función de administración de servicios
SMF :	Service Management Function : Función de administración de servicios

SMS :	Service Management System : Sistema de administración de servicios
SN :	Service Node : Nodo de servicio
SNMP :	Simple Network Management Protocol : Protocolo simple manejador de red
SPC :	Store Program Control : Control por programa almacenado
SRF :	Specialized Resource Function : Función de recursos especializados
SS7 :	Signaling System number 7 : Sistema de señalización número 7
SSCP :	Service Switching Control Point : Punto de conmutación y control de servicio
SSF :	Service Switching Function : Función de conmutación de servicios
SSN :	Subsystem Number : Numero de subsistema
SSP :	Service Switching Point : Punto de conmutación de servicio
STP :	Signaling Transfer Point : Punto de transferencia de señalización
TCAP :	Transaction Capabilities Application Part : Parte de aplicación de las capacidades de transacción
TUP :	Transaction User Part : Parte de la transacción de usuario
UDT :	Unit Data Transfer : Unidad de transferencia de información
UDTS :	Unit Data Transfer Service : Unidad de servicio de transferencia de información
UPT :	Universal Personal Telecommunication : Numero personal universal
VOT :	Televote : Televoto
VPN :	Virtual Private Network : Redes virtuales privadas