

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

Simulación de una Red VSAT *Full-Duplex* para acceso a Internet
usando la plataforma DVB-S y DVB-RCS

TOMO I

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES

HERNÁN VINICIO BARBA MOLINA

JUAN FRANCISCO CHAFLA ALTAMIRANO

DIRECTOR: ING. PABLO HIDALGO L.

Quito, Marzo 2006

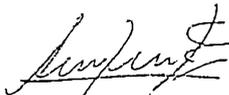
DECLARACIÓN

Nosotros, Hernán Vinicio Barba Molina, Juan Francisco Chafra Altamirano, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad intelectual, por su Reglamento y por la normatividad institucional vigente.



Hernán Vinicio Barba Molina



Juan Francisco Chafra Altamirano

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Hernán Vinicio Barba Molina y Juan Francisco Chafra Altamirano, bajo mi supervisión.



Ing. Pablo Hidalgo L.
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

En primer lugar, debo dar las gracias a Dios por darme la oportunidad de llegar a esta instancia en mi vida y permitirme escribir estas líneas.

Quiero expresar un profundo agradecimiento a mis padres por el apoyo incondicional que siempre me han brindado y por haber sido tan buenos y pacientes conmigo.

A mis hermanos por su ayuda y preocupación; realmente, ustedes han sido mi motivación y mi ejemplo. Un agradecimiento especial para mi hermano Gustavo por la valiosa ayuda prestada en la realización de este trabajo.

A mis amigos por compartir tantos momentos memorables e inolvidables. Gracias Glenda y Hernán por el apoyo y soporte brindado en todo este tiempo. A Xavier, Esteban, Carlos muchas gracias por su amistad que, pese a las circunstancias, aún perdura; siempre es un gusto volver a verlos y recordar esos días de universidad.

En realidad son muchas las personas que se merecen mi gratitud, aquellas que de una u otra forma viven mis triunfos con alegría y mis fracasos con tristeza, para todas ellas un agradecimiento infinito.

Juan

A Dios.

A mis padres.

Juan

AGRADECIMIENTO

A Dios por permitirme vivir día a día y colmarme de bendiciones.

A mis padres por su apoyo incondicional que me han brindado. Gracias a ellos hoy puedo ver la culminación de una etapa más del camino de mi vida.

A mi hermana por permitirme ser de ella más que un hermano, un amigo.

A mis grandes amigos Carlos, Javier, Esteban y Glenda; gracias por brindarme su amistad y estar siempre presentes en todo aquellos momentos importantes. Así mismo, y de manera especial, quiero agradecer infinitamente a Juan por ser mi amigo, confidente y ejemplo para mi persona.

A Gustavo Chafra por su constante preocupación y ayuda en la elaboración del presente trabajo.

Y a todas aquellas personas que de una u otra forma nos ayudaron y nos colaboraron.

Hernán

A mis padres

Hernán

CONTENIDO GENERAL

TOMO I

CONTENIDO GENERAL	I
ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABLAS	XVII
RESUMEN	XIX
PRESENTACIÓN	XXII

CAPÍTULO 1

Marco Teórico

1.1 Comunicaciones Satelitales Basadas en la Tecnología de Redes VSAT	1
1.1.1 Configuraciones de las Redes VSAT	5
1.1.1.1 Configuración en malla	7
1.1.1.2 Configuración en estrella	8
1.1.1.3 Comparación entre configuraciones malla y estrella	11
1.1.2 Estaciones Terrenas de las Redes VSAT	12
1.1.2.1 Estación VSAT	12
1.1.2.1.1 La unidad externa (ODU)	13
1.1.2.1.2 La unidad interna (IDU)	16
1.1.2.2 Estación <i>Hub</i>	17
1.1.2.2.1 Funciones operacionales	20
1.1.2.2.2 Funciones administrativas	21
1.1.3 Aplicaciones de las Redes VSAT y Tipos de Tráfico	21
1.1.3.1 Comunicaciones de datos	22
1.1.3.2 Comunicaciones de voz	22
1.1.3.3 Comunicaciones de vídeo	23
1.1.3.3.1 En el enlace <i>outbound</i>	23
1.1.3.3.2 En el enlace <i>inbound</i>	23

1.1.3.4	Redes VSAT para aplicaciones civiles	23
1.1.3.4.1	Tipos de servicio	23
1.1.3.4.2	Tipos de tráfico	25
1.1.3.5	Redes VSAT para aplicaciones militares	26
1.1.4	Ventajas y desventajas de una red VSAT	27
1.1.4.1	Ventajas	27
1.1.4.2	Desventajas	28
1.2	Redes Satelitales de Banda Ancha	29
1.2.1	El sistema satelital SKYPLEX	32
1.2.2	El sistema satelital IBIS	34
1.2.3	El sistema satélite SPACEMUX	37
1.3	Redes Satelitales VSAT Basadas en la Tecnología DVB-S/RCS	39
1.3.1	Redes Satelitales VSAT DVB Multimedia Interactivas	41
1.3.2	Ventajas y Desventajas Frente a Esquemas de Utilización Actuales	42
1.3.2.1	Ventajas	42
1.3.2.2	Desventajas	44
	Referencias Bibliográficas del Capítulo 1	45

CAPÍTULO 2

Estándares DVB-S/RCS

2.1	El sistema MPEG-2	47
2.1.1	Formación de los flujos de señal MPEG-2	47
2.1.2	Flujo de Programa PS (<i>Program Stream</i>)	49
2.1.3	Flujo de Transporte TS (<i>Transport Stream</i>)	49
2.1.3.1	Formación del Múltiplex de Transporte TS	51
2.1.3.2	Estructura del paquete de transporte MPEG-2 TS	52
2.1.3.2.1	Campo de adaptación del paquete MPEG-2	56
2.1.4	Información específica de los programas (PSI)	57
2.1.4.1	<i>Program Association Table</i> (PAT)	58
2.1.4.2	<i>Program Map Table</i> (PMT)	59
2.1.4.3	<i>Conditional Access Table</i> (CAT)	60
2.2	El estándar DVB-S	60

2.2.1	Sistema de transmisión DVB-S	61
2.2.1.1	Definición de sistema	61
2.2.1.2	Adaptación a las características del <i>transponder</i> satelital	63
2.2.1.3	Interfaces	64
2.2.1.4	Codificación de canal	64
2.2.1.4.1	Inversión del sincronismo y dispersión de energía	64
2.2.1.4.2	Codificación Reed-Solomon	65
2.2.1.4.3	Entrelazado (<i>interleaving</i>)	65
2.2.1.4.4	Codificación convolucional y código perforado	66
2.2.1.4.5	Filtrado banda base	66
2.2.1.4.6	Modulación QPSK	67
2.2.2	Información del Servicio (SI)	67
2.2.2.1	<i>Network Information Table</i> (NIT)	68
2.2.2.2	<i>Service Description Table</i> (SDT)	70
2.2.2.3	<i>Event Information Table</i> (EIT)	71
2.2.2.4	<i>Time and Data Table</i> (TDT)	71
2.2.2.5	<i>Bouquet Association Table</i> (BAT)	72
2.2.2.6	<i>Running Status Table</i> (RST)	72
2.2.2.7	<i>Time Offset Table</i> (TOT)	72
2.2.2.8	<i>Stuffing Table</i> (ST)	73
2.2.2.9	Códigos PID especificados por DVB-SI	73
2.2.2.10	Arquitectura de las Tablas de Información del Servicio (SI)	74
2.2.2.10.1	Distribución de las secciones en los paquetes de transporte	74
2.2.3	Especificaciones para la Difusión de Datos	75
2.2.3.1	Tubo de datos (<i>Data Pipe</i>)	76
2.2.3.2	Carrusel de datos (<i>Data carousel</i>)	76
2.2.3.3	Flujo continuo de datos (<i>Data streaming</i>)	77
2.2.3.4	Carrusel de objetos (<i>Object carousel</i>)	77
2.2.3.5	Encapsulamiento Multi-Protocolo (MPE)	77
2.2.3.5.1	Transporte de datos con MPE	78
2.2.3.5.2	Información en la SI	82
2.3	El Estándar DVB-RCS	82
2.3.1	Modelos de referencia para redes satelitales interactivas DVB	84

2.3.1.1	Modelo de pila de protocolos	84
2.3.1.2	Modelo genérico del sistema	86
2.3.1.3	Modelo de referencia para Redes Satelitales Interactivas	87
2.3.2	El Enlace de Ida o <i>Forward Link</i>	89
2.3.3	Especificaciones para la Capa Física para el Canal de Retorno	89
2.3.3.1	Sincronización de la RCST	90
2.3.3.1.1	Control de temporización	90
2.3.3.1.2	Sincronización de la portadora	91
2.3.3.1.3	Sincronización de la ráfaga	91
2.3.3.1.4	Sincronización del reloj de símbolo	92
2.3.3.2	Formato de ráfaga	92
2.3.3.2.1	Formato de la ráfaga de tráfico (TRF)	93
2.3.3.2.2	Formato de las ráfagas de sincronización (SYNC) y (ACQ)	94
2.3.3.2.3	Formato de la ráfaga de señalización del canal común (CSC)	96
2.3.3.3	Aleatorización para dispersión de energía	98
2.3.3.4	Codificación	98
2.3.3.5	Modulación	98
2.3.3.6	Envío de mensajes MAC	100
2.3.3.6.1	Métodos basados en el campo SAC	101
2.3.3.6.2	Método DULM para la entrega de mensajes MAC	104
2.3.3.7	Acceso múltiple MF-TDMA	108
2.3.3.7.1	MF-TDMA fijo	108
2.3.3.7.2	MF-TDMA dinámico	109
2.3.3.8	Segmentación de la capacidad del enlace de retorno	110
2.3.3.8.1	Supertramas	110
2.3.3.8.2	Tramas	111
2.3.3.8.3	Ranuras de tiempo	112
2.3.3.9	Categorías de Peticiones de Capacidad	113
2.3.3.9.1	Asignación de tasa continua CRA	114
2.3.3.9.2	Asignación de capacidad basada en tasa RBDC	114
2.3.3.9.3	Asignación de capacidad basada en volumen VBDC y AVBDC	115
2.3.3.9.4	Asignación de capacidad disponible FCA	116
2.3.4	Múltiples Canales por Terminal	117

2.3.5	Procedimientos de Sincronización	119
2.3.5.1	Secuencia global de eventos	119
2.3.5.2	Procedimiento inicial de sincronización	123
2.3.5.3	Procedimiento de entrada (<i>Logon</i>)	124
2.3.5.4	Procedimiento de sincronización gruesa (Opcional)	125
2.3.5.5	Procedimiento de sincronización fina (Opcional)	126
2.3.5.6	Procedimiento para mantener la sincronización	127
2.3.5.7	Procedimiento de salida (<i>logoff</i>)	127
2.3.5.7.1	Normal	128
2.3.5.7.2	Anormal	128
2.3.6	Pila de Protocolos	128
2.3.6.1	RCST Tipo A (IP)	129
2.3.6.2	RCST Opcional Tipo B (ATM nativo)	131
2.3.7	Direccionamiento de las RCSTs	131
2.3.8	Señalización del Enlace de Ida o FLS (<i>Forward Link Signalling</i>)	133
2.3.8.1	<i>Superframe Composition Table</i> (SCT)	135
2.3.8.2	<i>Frame Composition Table</i> (FCT)	135
2.3.8.3	<i>Time-Slot Composition Table</i> (TCT)	135
2.3.8.4	<i>Satellite Position Table</i> (SPT)	136
2.3.8.5	<i>Correction Message Table</i> (CMT)	136
2.3.8.6	<i>Terminal Burst Time Plan</i> (TBTP)	136
2.3.8.7	<i>Terminal Information Message</i> (TIM)	137
2.3.9	Acceso al <i>Forward Link Signaling</i> (FLS)	138
2.3.9.1	Sincronización física del enlace	139
2.3.9.2	Localización del enlace de ida	139
2.3.9.3	Sincronización del NCR	140
2.3.9.4	Carga de las tablas DVB-RCS específicas	142
2.3.10	Señalización del Canal de Retorno	142
	Referencias Bibliográficas del Capítulo 2	144

CAPÍTULO 3

Network Simulator

3.1	Principios del NS	145
3.1.1	Lenguaje de Usuario Otcl	150
3.1.2	Ejemplo de Simulación Sencillo	153
3.1.3	Componentes de Red	159
3.1.3.1	Nodos y ruteo	160
3.1.3.2	Enlaces	161
3.1.3.2.1	Trazo	162
3.1.3.2.2	Control de cola	163
3.1.3.3	Ejemplo de flujo de paquete	163
3.1.4	Paquete	164
3.1.5	Archivos de Trazo	166
3.2	Manipulación de los Archivos de Trazo	168
3.2.1	Procesamiento de Archivos con AWK	168
3.2.2	Uso de GREP	170
3.2.3	Procesamiento de Archivos de Datos con PERL	171
3.2.4	Graficación con GNUPLOT	173
3.2.5	Graficación con XGRAPH	176
3.3	Variables Aleatorias	177
3.3.1	Semillas y Generadores	177
3.3.2	Creación de Variables Aleatorias	178
3.4	Extensiones a NS	181
3.4.1	Encadenamiento Otcl	183
3.4.1.1	Exportar una clase C++ a Otcl	184
3.4.1.2	Exportar variables de una clase C++ a Otcl	185
3.4.1.3	Exportar comando de control de objeto de C++ a Otcl	186
3.4.1.4	Ejecución de un comando Otcl desde C++	187
3.5	Network Animator (NAM)	188
	Referencias Bibliográficas del Capítulo 3	193

TOMO II

CAPÍTULO 4

Modelo de Simulación de Red

4.1	Arquitectura de Red	194
4.1.1	Consideraciones Específicas a la Red DVB-S/RCS	196
4.1.1.1	Único NCC para toda la red	196
4.1.1.2	Múltiples alimentadores y pasarelas DVB	196
4.1.1.3	Único satélite GEO	197
4.1.2	Propuesta de Arquitectura de Red	197
4.1.3	Aspectos Generales de Diseño	200
4.1.3.1	Elección del tipo de estación RCST	200
4.1.3.2	El NCC o <i>hub</i> central	203
4.1.3.3	Organización del recurso satelital	204
4.1.3.4	Direccionamiento	205
4.1.3.5	Determinación del estado de las RCSTs	206
4.1.3.6	Asignación de los identificadores de paquete PID	207
4.1.3.7	Problema del ensamblado MPE	208
4.1.3.8	Envío de mensajes de señalización	211
4.1.3.8.1	Mensajes de señalización desde la RCST al NCC	212
4.1.3.8.2	Mensajes desde el NCC a la estación RCST	214
4.1.3.8.3	Petición y asignación de recursos satelitales RCS	214
4.2	Pila de protocolos	216
4.2.1	Estaciones RCST como <i>Routers</i> IP	216
4.2.2	Interconexión	216
4.2.3	Arquitectura de protocolos de la red RCS	217
4.2.3.1	Plano de datos y control de las estaciones de usuario	217
4.2.3.2	Plano de datos y control de las estaciones RCST	218
4.2.3.3	Plano de datos y control del servidor	219
4.2.3.4	Plano de control del NCC	220
4.3	Asignación de recursos	220

4.3.1	Asignación Inicial de Recursos del Servidor	221
4.3.2	Asignación Dinámica de Recursos a las Estaciones RCST	222
4.4	Control de conexión	225
4.4.1	Establecimiento de Conexión entre Cliente y Servidor	225
4.5	Modelo de simulación	226
4.5.1	Validación y Verificación del Modelo	226
4.5.1.1	Técnicas de validación	227
4.5.1.2	Técnicas de verificación	227
4.5.2	Intervalos de Confianza	228
4.5.3	Eliminación del Transitorio	229
4.5.4	Generación de Números Aleatorios	230
4.5.5	Implantación de la Arquitectura de Red	231
4.5.5.1	Topología NS de la arquitectura de red	235
4.5.6	Descripción de Modelo de Red	236
4.5.6.1	Estructura de la multitrama MF-TDMA	237
4.5.6.2	Tamaño de la supertrama	240
4.5.6.3	Tipos y configuración de portadoras	241
4.5.6.4	Ráfagas de señalización CSC y CSYNC	242
4.5.6.5	Ráfagas de señalización SYNC	243
4.5.6.6	Asignación de recursos basada en volumen VBDC	245
4.5.6.7	Asignación de capacidad disponible FCA	247
4.5.6.8	Validación de la asignación de recursos TDMA	247
4.6	Modelos de tráfico de Internet	250
4.6.1	Navegación Web	250
4.6.1.1	Modelo de tráfico para servidor Web	252
4.6.1.2	Modelo de tráfico de usuario-cliente Web	253
4.6.1.3	Modelo UMTS de tráfico Web	255
4.6.1.4	Modelo WWW para redes inalámbricas (Único usuario)	259
4.6.2	Correo Electrónico e-mail	262
4.6.2.1	Modelo SMTP	263
4.6.2.2	Modelo POP3/IMAP	265
4.6.3	Servicios Remotos Interactivos (IRS) (TELNET/SLOGIN)	265

4.6.4 Servicios de Transferencia de Archivos FTP	266
4.7 Análisis Comparativo de los Modelos de Tráfico	267
4.8 Escenario de Simulación	268
4.8.1 Navegación Web (<i>Web surfing</i>)	269
Referencias Bibliográficas del Capítulo 4	271

CAPÍTULO 5

Simulación y Análisis de Resultados

5.1 Consideraciones Generales	274
5.1.1 Consideraciones del Escenario de Simulación	274
5.1.2 Consideraciones de la Topología de Red	276
5.1.3 Consideraciones del Entorno de Simulación	277
5.1.4 Consideraciones de los Archivos de Trazo	279
5.1.4.1 Archivos de Trazo Generales	280
5.1.4.1.1 Archivo de Trazo General <i>out.tr</i>	281
5.1.4.1.2 Archivo de Trazo General <i>graph.txt</i>	282
5.1.4.1.3 Archivo de Trazo General <i>rsc.out</i>	283
5.1.4.2 Archivos de Trazo Específicos	287
5.1.4.2.1 Archivos de Trazo Específicos de <i>Throughput</i>	288
5.1.4.2.2 Archivos de Trazo Específicos de Retardos	288
5.1.4.3 Archivos de Trazo de Resultados	291
5.2 Evaluación de la Arquitectura de Red	292
5.3 Análisis de Sensibilidad y Saturación	307
5.4 Análisis de la Factibilidad de la Implementación del Sistema	320
Referencias Bibliográficas del Capítulo 5	324

CAPÍTULO 6

Conclusiones y Recomendaciones

6.1 Conclusiones	325
6.2 Recomendaciones	334

Lista de Acrónimos	336
---------------------------	------------

Anexo A

Instalación de NS

A.1 Consideraciones para la instalación de <i>Linux</i>	340
A.2 Instalación de NS	340
A.2.1 Obtención del <i>software</i> gratuito	340
A.2.2 Configuración de NS en <i>Linux</i>	342
A.3 Explore2fs	343

Anexo B

Manual del Usuario

B.1 Iniciación de una simulación	344
B.1.1 Configuración para un grupo de simulaciones	345
B.1.2 Configuración de parámetros de simulación	346
B.2 Formato de los archivos de trazo	347
B.2.1 Archivos de trazo específicos	347
B.3 Mensajes del archivo <i>res.out</i>	350

Anexo C

Estructuras de los Programas

<i>residencial.tcl</i>	366
<i>init.tcl</i>	368
<i>gentop.tcl</i>	369
<i>trafficgen.tcl</i>	372
<i>ncc.tcl</i>	374
<i>constants.tcl</i>	375
<i>rv_source.tcl</i>	376
<i>scenario.tcl</i>	377
<i>throughput.tcl</i>	378

rds.cc	379
RCST.cc	380
server.cc	388
ncc.cc	389
calc_test.c (a.out)	397
promedio_retard.awk	399
promedio_retard2.awk	400
promedio_thr.awk	401
promedio_thr2.awk	402
Referencias Bibliográficas	403

Índice de Figuras

Figura 1.1 VSAT un paso hacia la reducción de estaciones terrenas.	2
Figura 1.2 Estaciones Troncales.	3
Figura 1.3 Desde estaciones troncales hasta VSATs.	4
Figura 1.4 Definición de <i>uplink</i> y <i>downlink</i> .	6
Figura 1.5 Satélite Geoestacionario.	6
Figura 1.6 Configuración en malla.	8
Figura 1.7 Configuración en estrella bidireccional.	9
Figura 1.8 Configuración en estrella unidireccional.	10
Figura 1.9 Conectividad del usuario terminal dentro de una red VSAT malla.	11
Figura 1.10 Conectividad del usuario terminal usando el <i>hub</i> como un relevador en una red VSAT estrella.	12
Figura 1.11 Diagrama de bloques de una estación terrena.	13
Figura 1.12 Fotografía de una ODU de una estación VSAT.	13
Figura 1.13 Fotografía de un contenedor electrónico.	15
Figura 1.14 Vista frontal de una IDU de una estación VSAT.	17
Figura 1.15 Fotografía de una estación <i>hub</i> .	19
Figura 1.16 Subsistemas del <i>hub</i> .	19
Figura 1.17 Esquema general de un sistema satelital de servicios múltiples.	31
Figura 1.18 Arquitectura del sistema Skyplex.	32
Figura 1.19 Arquitectura de multiplexación del sistema Skyplex.	33
Figura 1.20 Concepto del sistema IBIS de Alcatel.	35
Figura 1.21 Arquitectura general del sistema SpaceMux.	37
Figura 1.22 Principio de funcionamiento de una red ICMSN.	41
Figura 2.1 Esquema conceptual de la formación de flujos de señal MPEG-2.	48
Figura 2.2 División de un PES en una serie de paquetes MPEG-2 TS.	50
Figura 2.3 Conformación del Múltiplex de Transporte TS.	51
Figura 2.4 Paquete de transporte MPEG-2.	52
Figura 2.5 Detalle del campo de adaptación del paquete MPEG-2.	56
Figura 2.6 Forma de una Tabla PAT.	58
Figura 2.7 Estructura de una Tabla PMT.	59

Figura 2.8 Diagrama de bloques funcional del Sistema de Transmisión DVB-S. _____	62
Figura 2.9 Ejemplo de Transferencia de Servicios a un nuevo Sistema de Entrega. ____	70
Figura 2.10 Primer comienzo de sección dentro de un paquete TS. _____	75
Figura 2.11 Organización de la dirección MAC en campos de sección MPE. _____	80
Figura 2.12 Estructura en capas de un Modelo de Referencia para un Sistema Genérico DVB-S/RCS. _____	85
Figura 2.13 Modelo de Referencia de un Sistema Genérico para Sistemas Interactivos. _	87
Figura 2.14 Modelo de referencia para Redes Satelitales Interactivas. _____	89
Figura 2.15 Diagrama de Bloques del procesamiento de señal en banda base del canal de retorno en un transmisor RCST. _____	90
Figura 2.16 Composición de un ráfaga ATM. _____	93
Figura 2.17 Composición de una ráfaga MPEG2-TS. _____	93
Figura 2.18 Composición de una ráfaga SYNC. _____	95
Figura 2.19 Composición de la ráfaga ACQ. _____	95
Figura 2.20 Composición de una ráfaga CSC. _____	96
Figura 2.21 Proceso de Modulación. _____	99
Figura 2.22 Asignación de bits dentro de la constelación QPSK. _____	99
Figura 2.23 Formato del Mensaje DULM con el formato ATM. _____	105
Figura 2.24 DULM sobre paquetes MPEG. _____	106
Figura 2.25 MF-TDMA de ranuras fijas. _____	108
Figura 2.26 MF-TDMA de ranuras dinámicas, usando una RCST flexible. _____	109
Figura 2.27 Ejemplo típico de supertramas de una Red Satelital Interactiva. _____	110
Figura 2.28. Ejemplo de composición de una supertrama. _____	112
Figura 2.29 Ejemplo de composición de una trama. _____	113
Figura 2.30 Representación funcional del uso de los canales. _____	117
Figura 2.31 Diagrama del estado del procedimiento de entrada y sincronización de una RCST. _____	121
Figura 2.32 Ejemplo del flujo de señalización de entrada entre el NCC y la RCST. ____	122
Figura 2.33 Procedimiento de entrada. _____	124
Figura 2.34 Procedimiento de sincronización gruesa. _____	125
Figura 2.35 Procedimiento de sincronización fina. _____	126
Figura 2.36 Procedimiento para mantener la sincronización. _____	127

Figura 2.37 Ejemplo de una pila de protocolos para tráfico de usuario con un RCST Tipo A (IP/AAL5/ATM/MPEG2/DVBS es opcional en el enlace de ida).	130
Figura 2.38 Pila de protocolos para señalización.	130
Figura 2.39 <i>Stack</i> de Protocolos para el FLS.	134
Figura 2.40 Ejemplo de localización del enlace de ida.	141
Figura 3.1 Esquema de simulación en NS.	146
Figura 3.2 Dualidad C++ y Otcl.	148
Figure 3.3 Arquitectura de NS.	149
Figura 3.4 Una topología de red simple y un escenario de simulación.	153
Figura 3.5 Jerarquía de clases (parcial).	159
Figura 3.6 Nodo (<i>Unicast</i> y <i>Multicast</i>).	160
Figura 3.7 Enlace.	161
Figura 3.8 Objetos de trazo insertados.	162
Figura 3.9 Control de cola.	163
Figura 3.10 Ejemplo de flujo de paquete.	164
Figura 3.11 Formato de un paquete NS.	165
Figura 3.12 Ejemplo de formato de trazo.	167
Figura 3.13 Ejemplo de GNU PLOT.	176
Figura 3.14 Ejemplo de uso del XGraph.	177
Figura 3.15 Estructura del directorio NS.	182
Figura 3.16 Visualización en el NAM del ejemplo 3.15 a los 6.6 seg.	191
Figura 4.1 Esquema general de la arquitectura de red.	197
Figura 4.2 Esquema de la red satelital para acceso a Internet.	203
Figura 4.3 Encapsulamiento IP sobre MPE/MPEG-2.	209
Figura 4.4 Intercalado de secciones MPE con un mismo PID.	210
Figura 4.5 Formato DULM sobre paquetes MPEG-2.	214
Figura 4.6 Planos de datos y control de las estaciones de usuario.	217
Figura 4.7 Planos de datos y control de las RCSTs.	218
Figura 4.8. Planos de datos y control del servidor.	219
Figura 4.9. Plano de control del NCC.	220
Figura 4.10 Adquisición inicial de recursos del servidor.	221

Figura 4.11 Composición del campo SAC. _____	223
Figura 4.12 Petición de recursos DVB-RCS por parte de las RCSTs. _____	224
Figura 4.13 Conexión cliente – servidor. _____	226
Figura 4.14 Jerarquía de clases RCS NS. _____	232
Figura 4.15 Estructura interna de un nodo RCS NS. _____	233
Figura 4.16 Modelo NS de la arquitectura de red. _____	235
Figura 4.17 Validación del envío de ráfagas de sincronización SYNC. _____	245
Figura 4.18 Retardo medio de paquete (trama 70,311 ms). _____	249
Figura 4.19 Retardo medio de paquete (trama 46.87 ms). _____	249
Figura 4.20 Tiempos medios inter-peticiones. _____	252
Figura 4.21 Característica típica de una sesión de servicio de paquete. _____	256
Figura 4.22 Fases de actividad WWW. _____	260
Figura 5.1 Topología básica de la Red para Acceso a Internet DVB-S/RCS. _____	276
Figura 5.2 Ejemplo de un archivo <i>out.tr</i> . _____	281
Figura 5.3 Visualización NAM de <i>graph.txt</i> . _____	283
Figura 5.4 Esquema de obtención de los archivos de <i>throughput</i> . _____	289
Figura 5.5 Esquema de obtención de los archivos de retardos. _____	290
Figura 5.6 Esquema de obtención de los archivos de resultados. _____	291
Figura 5.7 Retardo de tráfico BE desde el ISP (FCA - activado). _____	294
Figura 5.8 Retardo de tráfico BE desde el RCST (FCA - activado). _____	295
Figura 5.9 <i>Throughput</i> de tráfico BE desde el ISP (FCA - activado). _____	296
Figura 5.10 <i>Throughput</i> de tráfico BE desde el RCST (FCA - activado). _____	296
Figura 5.11 Retardo de tráfico BE desde el RCST (FCA - desactivado). _____	300
Figura 5.12 <i>Throughput</i> de tráfico BE desde el ISP (FCA - desactivado). _____	301
Figura 5.13 <i>Throughput</i> de tráfico BE desde el RCST (FCA - desactivado). _____	302
Figura 5.14 Comparación entre el <i>throughput</i> de un RCST y el del RCS_SAT. _____	305
Figura 5.15 Comparación entre el <i>throughput</i> del ISP el del RCS_SAT. _____	306
Figura 5.16 Tráfico BE del ISP para un ambiente con 16 RCST y otro con 18 RCST. _____	310
Figura 5.17 Retardo de tráfico BE desde el ISP (18 RCST). _____	311
Figura 5.18 Retardo Tráfico BE desde el RCST vs. No. de Usuarios. _____	312
Figura 5.19 Retardo Tráfico BE desde el ISP vs. No. de Usuarios. _____	313
Figura 5.20 Tráfico BE DVB-RCS vs. No. de Usuarios. _____	314

Figura 5.21 Tráfico BE ISP vs. No. de Usuarios. _____	315
Figura 5.22 Utilización de los Mensajes de Señalización MAC. _____	316
Figura 5.23 Caudal reservado vs. Caudal cursado. _____	317
Figura 5.24 Tráfico BE del ISP para un ambiente con 26 RCST y otro con 28 RCST. _	319
Figura A.1 Explore2fs. _____	343
Figura B.1 Formato del nombre del archivo de trazo específico de <i>throughput</i> . _____	347
Figura B.2 Archivo de Trazo General rcs.out. _____	348
Figura B.3 Formato del nombre del archivo de trazo de retardo. _____	349

Índice de Tablas

Tabla 1.1 Valores típicos para las partes de la ODU de una estación VSAT. _____	16
Tabla 1.2. Valores típicos para las partes de la ODU de una estación <i>hub</i> . _____	18
Tabla 2.1 Estructura de la cabecera de un Paquete MPEG-2 TS. _____	52
Tabla 2.2 Tabla general de PIDs. _____	54
Tabla 2.3 Valores del control de enmascaramiento de transporte. _____	55
Tabla 2.4 Valor del control del campo de adaptación. _____	55
Tabla 2.5 Constitución del campo de adaptación del paquete MPEG-2. _____	57
Tabla 2.6 Interfaces del sistema. _____	64
Tabla 2.7 Valores de PID para las Tablas SI. _____	73
Tabla 2.8 Sintaxis de la Sección de Datagrama MPE. _____	79
Tabla 2.9 Parámetros del campo de datos de la ráfaga CSC. _____	97
Tabla 2.10 Campo <i>RCST Capability</i> de la Ráfaga CSC. _____	97
Tabla 2.11 Sintaxis del campo SAC. _____	101
Tabla 2.12 <i>Scaling_Factor</i> . _____	102
Tabla 2.13 <i>Capacity_Request_Type</i> . _____	102
Tabla 2.14 <i>M_and_C_Message</i> . _____	103
Tabla 2.15 Tipos de IE en MPEG. _____	107
Tabla 2.16 Distribución de los PID y <i>Table_id</i> para las tablas SI. _____	134
Tabla 4.1 Tipos de terminales interactivos por satélite. _____	200
Tabla 4.2 Comparación de eficiencia entre MPEG2/MPE y ATM/AAL5. _____	202
Tabla 4.3 Estructura de trama para una tasa de símbolo de 270 ksymb/s. _____	238
Tabla 4.4 Tasas de usuario (Tasa de símbolo 270 ksymb/s). _____	240
Tabla 4.5 Tipos de portadoras y tasas de bits por portadora. _____	242
Tabla 4.6 Parámetros del modelo y sus correspondientes distribuciones. _____	254
Tabla 4.7 Parámetros del modelo considerando las fuentes ON/OFF. _____	254
Tabla 4.8 Parámetros del modelo con escenarios corporativos y educacionales. _____	255
Tabla 4.9 Valores medios definidos para una distribución típica del servicio WWW. _____	259
Tabla 4.10 Variables aleatorias que describen el modelo. _____	261
Tabla 4.11 Distribuciones para las variables aleatorias de la tabla 4.10. _____	262

Tabla 4.12 Número de sesiones promedio para <i>Web Surfing</i> .	268
Tabla 4.13 Velocidad de generación de sesión para un número promedio y pico de sesiones.	270
Tabla 5.1 Parámetros de tráfico para el escenario residencial (17:00 – 24:00).	275
Tabla 5.2 Parámetros de configuración de la red.	279
Tabla 5.3 Prestaciones del escenario residencial (1 RCST - FCA activado).	293
Tabla 5.4 Prestaciones del escenario residencial (1 RCST - FCA desactivado).	299
Tabla 5.5 Prestaciones del escenario residencial (6 RCST).	304
Tabla 5.6 Prestaciones del escenario residencial (8 RCST – 240 usuarios).	307
Tabla 5.7 Prestaciones del escenario residencial (10 RCST – 300 usuarios).	308
Tabla 5.8 Prestaciones del escenario residencial (16 RCST – 480 usuarios).	309
Tabla 5.9 Prestaciones del escenario residencial (18 RCST – 540 usuarios).	309
Tabla 5.10 Retardos Tráfico BE RCST/ISP vs. No. de usuarios.	312
Tabla 5.11 <i>Throughput</i> Tráfico BE vs. No. de usuarios.	314
Tabla 5.12 Porcentajes de utilización de los mensajes MAC.	315
Tabla 5.13 Retardos Tráfico BE vs. No. de usuarios (fuera de la hora pico).	318
Tabla 5.14 <i>Throughput</i> Tráfico BE vs. No. de usuarios (fuera de la hora pico).	318
Tabla 5.15 Consideraciones de diseño de un ISP.	321
Tabla B.1 Variables configurables en <i>init.tcl</i> .	346
Tabla B.2 Sintaxis del nombre de los archivos de <i>throughput</i>	349
Tabla B.3 Sintaxis del nombre de los archivos de retardo	350

RESUMEN

Las necesidades y comportamiento de los usuarios de las redes de comunicaciones han evolucionado a la par que lo ha hecho la tecnología. El crecimiento exponencial que ha experimentado la demanda de servicios tales como el Internet, ha motivado la inclusión de cierta interactividad en redes que inicialmente fueron concebidas para la difusión. Éste es el caso de las redes VSAT basadas en el sistema DVB-S (*Digital Video Broadcasting by Satellite*) de amplia aceptación en el mercado europeo.

El esquema tradicional de acceso interactivo adoptado por las redes VSAT DVB-S cuenta con un canal de retorno provisto por un operador terrestre generalmente distinto al proveedor satelital. Este tipo de esquema ha hecho que las redes VSAT sean dependientes de la infraestructura terrestre instalada; por tanto, las ventajas inherentes a las comunicaciones satelitales se han visto limitadas.

Es precisamente, por la necesidad de aprovechar todo el potencial de las comunicaciones satelitales, que surge el estándar DVB-RCS (*Digital Video Broadcasting - Return Channel System*), el cual provee las pautas para la creación de redes totalmente interactivas por satélite.

En este proyecto se identifica entonces la tendencia tecnológica de las redes VSAT a la utilización de las normas DVB, las cuales permiten proveer a sus usuarios con servicios interactivos prescindiendo de un canal de retorno terrestre en un ambiente totalmente satelital.

Surge entonces la necesidad de estudiar el comportamiento de una red de acceso a Internet sobre la plataforma DVB-S/RCS con el fin de analizar la factibilidad de su implementación en nuestro medio. Puesto que en el País no se cuenta con un sistema de este tipo implementado, y debido a que su instalación

sería extremadamente costosa, este trabajo presenta la alternativa de modelar la arquitectura de una red DVB y el comportamiento de usuario Web, y simularlos a través del paquete NS (*Network Simulator*).

Para la consecución de los objetivos planteados en este proyecto, se ha partido primeramente de una revisión del estado del arte de la comunicaciones satelitales VSAT identificando su evolución, las ventajas y desventajas de su implementación y sobre todo la orientación que éstas muestran a la utilización de los estándares DVB. Se revisan también algunos sistemas VSAT de banda ancha actuales y propuestos para un futuro que incorporan la tecnología DVB para proveer a sus usuarios con servicios interactivos.

Posteriormente se realiza un estudio de los estándares DVB-S y DVB-RCS, destacando aquellos conceptos y recomendaciones de trascendencia para la elaboración del modelo de simulación de red. Se profundiza en conceptos claves tales como el Múltiplex de Transporte MPEG-2, los mecanismos de envío de mensajes de señalización, etc.

El simulador de redes NS es un aspecto importante tratado en el presente proyecto. De él se explica la mecánica de su funcionamiento, su estructura, sus definiciones y algunos ejemplos sencillos de simulaciones que permiten entender cómo trabaja dicho *software*. Además, la información incluida sobre NS constituye una base teórica que facilita la comprensión de cómo fue realizada la implantación del modelo de red en el simulador.

También, se describe la arquitectura de red detallando todas las consideraciones específicas y generales que han sido tomadas, así como los aspectos de diseño más relevantes. Posteriormente se elabora un modelo de simulación para la arquitectura propuesta, el cual incluye un entorno de aplicación del tipo residencial. Para la elaboración del modelo de simulación se han tomado en cuenta las propuestas de los sistemas de banda ancha DVB interactivos antes mencionados, así como también lo expuesto por la norma para la creación de sistemas DVB-S/RCS.

Se presentan también varios modelos que caracterizan el tráfico de Internet, todos ellos basados en medidas de entornos reales, de los cuales se escoge uno para alimentar al modelo de simulación.

El modelo de simulación ha sido sometido a varias pruebas de verificación y validación para comprobar la correcta implementación del mismo en el simulador. Esto se logró gracias a la utilización de algunos modelos teóricos que, bajo determinadas circunstancias, describen el comportamiento del sistema y cuya comparación con los resultados obtenidos en las simulaciones verificaron el correcto funcionamiento del modelo.

A continuación, se presenta la realización de las simulaciones del modelo de red para la consecución de resultados. Con dichos resultados obtenidos se realiza una evaluación de las prestaciones y funcionalidad de la plataforma DVB en el escenario determinado con el modelo de tráfico de usuario establecido (*Web surfing*).

Finalmente se comparan las prestaciones, obtenidas a través de las simulaciones, con las de un entorno real; verificando las ventajas que la arquitectura propuesta presenta frente a redes de utilización actuales, haciéndola competitiva no solo en áreas carentes de infraestructura terrestre sino también en zonas densamente pobladas como una ciudad.

En conclusión, el principal aporte del presente proyecto es la elaboración del modelo de simulación de la arquitectura de red propuesta, que junto con la implementación del escenario de simulación y el modelo de tráfico de usuario han permitido evaluar las prestaciones y funcionalidad de la "Red de Acceso a Internet sobre la plataforma DVB-S/RCS" planteada. Todos estos aspectos han permitido estudiar la factibilidad de la implementación de este sistema en un ambiente como el nuestro.

PRESENTACIÓN

El presente proyecto contiene las pautas para la creación de un sistema de red totalmente interactivo vía satélite basado en los estándares de difusión de vídeo DVB-S/RCS. Además, presenta un modelo de la arquitectura de red, implementado en el simulador de redes NS, del cual se realiza un análisis de prestaciones de la tecnología DVB a través de la consecución de varias simulaciones del sistema.

Es por eso que, la información detallada en este documento está dirigida, principalmente, a aquellas personas de la industria de las telecomunicaciones que deseen conocer de forma detallada las características de la plataforma DVB-S/RCS, las prestaciones de la arquitectura propuesta y la factibilidad de su implementación.

Para el logro de los objetivos planteados en este proyecto se ha recopilado una cantidad valiosa de información, adicional a los estándares, que puede ser relevante para otros campos. Por ejemplo, se detalla el manejo del simulador de redes NS, el cual puede convertirse en una excelente herramienta de aprendizaje para los estudiantes de ingeniería y personas en general con conocimientos básicos de redes de información.

De igual manera este trabajo puede ser considerado como una base para proyectos de investigación posteriores, ya que el alcance del tema propuesto es amplio. Es así que han sido planteadas también algunas líneas de investigación futuras, que con la ayuda de los resultados alcanzados en el presente proyecto, pueden aportar mayor información acerca del comportamiento del sistema DVB bajo otro escenario de simulación, por ejemplo.

CAPÍTULO 1

MARCO TEÓRICO

En este capítulo se pretende describir el estado del arte de las comunicaciones satelitales VSAT, dando una breve exposición de sus conceptos básicos y las alternativas que actualmente éstas ofrecen para acceder a servicios interactivos como el Internet, y analizando su tendencia futura.

En primera instancia se realiza una descripción de las comunicaciones satelitales basadas en la tecnología de redes VSAT; se revisan sus conceptos, definiciones básicas y aplicaciones, así como también las ventajas y desventajas de su implementación.

Seguidamente, se aborda el estudio de las redes de banda ancha, analizando sus conceptos y características más relevantes. De forma más particular, se tratan las comunicaciones satelitales basadas en los conceptos anteriores, describiendo la situación actual de éstas y las propuestas para un futuro.

Finalmente, y una vez identificados los puntos anteriores, se exponen las redes satelitales VSAT basadas en la tecnología DVB (*Digital Video Broadcasting*), prestando especial atención al estándar DVB-RCS (*Digital Video Broadcasting Return Channel System*) el cual establece las pautas para la construcción de sistemas satelitales multimedia completamente interactivos.

1.1 COMUNICACIONES SATELITALES BASADAS EN LA TECNOLOGÍA DE REDES VSAT [1]

Las VSATs (*Very Small Aperture Terminal*), terminales de muy pequeña abertura, son un paso intermedio en la búsqueda de la reducción del tamaño de las estaciones terrenas observado en las comunicaciones satelitales desde el lanzamiento de los primeros satélites a mediados de los años sesenta.

De hecho, las estaciones terrenas han evolucionado desde antenas de 30 metros de ancho, hasta estaciones con antenas de 60 centímetros, para la recepción directa de televisión transmitida por satélites de difusión, o terminales de mano para radiolocalización como los receptores GPS. Hoy en día inclusive, IRIDIUM y GLOBALSTAR ofrecen teléfonos satelitales de tamaño muy reducido. La figura 1.1 ilustra esta tendencia.

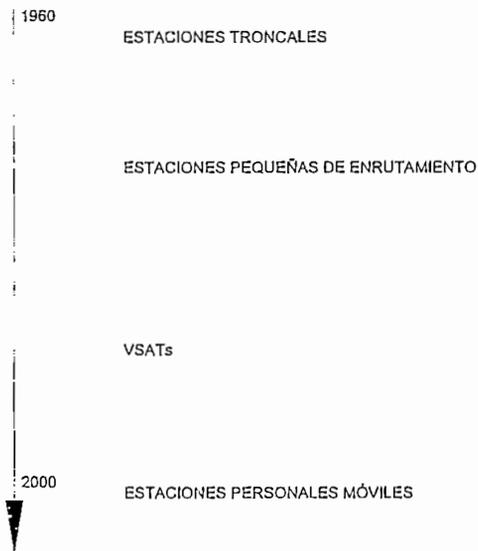


Figura 1.1 VSAT un paso hacia la reducción de estaciones terrenas.

Como se puede observar, las comunicaciones basadas en la tecnología VSAT alcanzan el nivel más bajo en una línea de productos que ofrece una gran variedad de servicios de comunicaciones. En el nivel superior de esta línea, se encuentran grandes estaciones llamadas Estaciones Troncales, las cuales están diseñadas para soportar enlaces satelitales de alta capacidad. Estas estaciones son usadas principalmente a través de redes conmutadas internacionales para soportar servicios telefónicos entre países, posiblemente ubicados en diferentes continentes.

La figura 1.2, ilustra cómo las estaciones troncales recogen el tráfico generado por los usuarios finales a través de enlaces terrestres que son parte de la red pública conmutada de un país. Estas estaciones son bastante costosas y requieren trabajos civiles considerables para su instalación.

Las capacidades de sus enlaces están en el rango de unos cuantos miles de canales telefónicos, o equivalentemente, alrededor de 100 Mbps. Éstos son controlados y administrados por operadores nacionales o por grandes compañías privadas de administración.

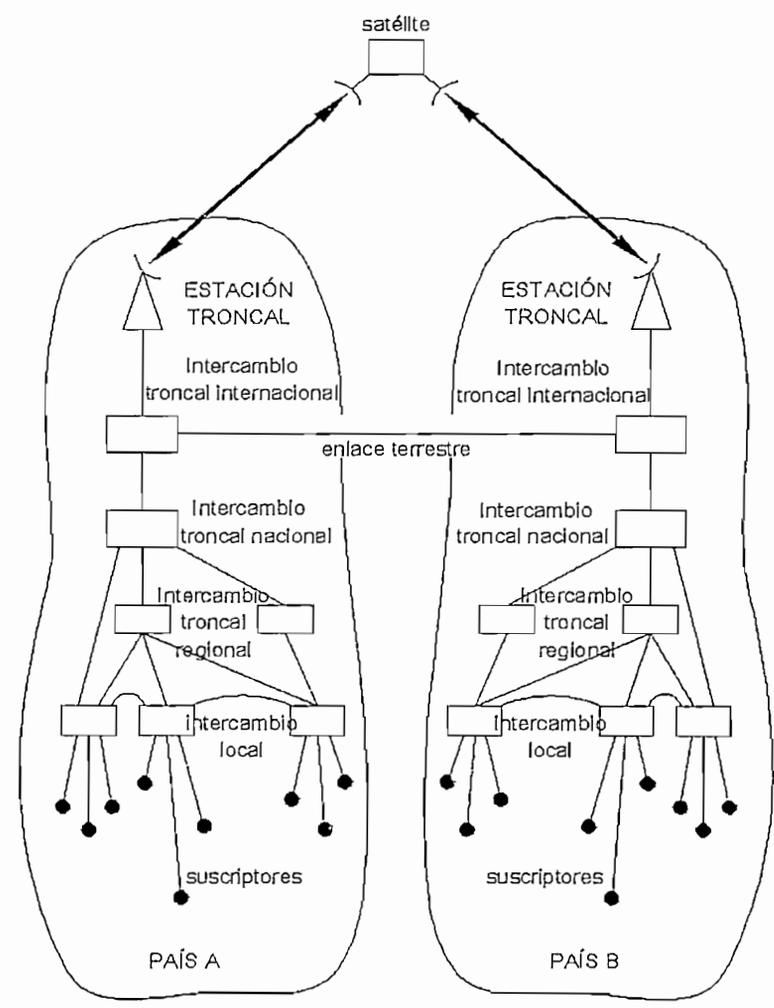


Figura 1.2 Estaciones Troncales.

Las VSAT por su parte son estaciones pequeñas con diámetros de antena menores a 2.4 metros, de ahí el nombre de "terminal de apertura pequeña". Tales estaciones no pueden soportar enlaces satelitales de gran capacidad, pero en cambio son baratas, con costos en el rango de 1000 a 5000 dólares por estación; son fáciles de instalar en lugares como la terraza de un edificio, el jardín de una casa, etc. Además, sus capacidades están en el orden de unas pocas decenas de Kbps, típicamente 56 o 64 Kbps.

El bajo costo de las VSAT las ha hecho muy populares, con un crecimiento de mercado en el orden de 20 a 25% anual en la década de los noventa. En esa época, había aproximadamente 50000 estaciones VSAT en funcionamiento a nivel mundial, y más de 600000, doce años después. Es probable que esta tendencia continúe.

Refiriéndose al transporte de la información, las estaciones VSAT son el equivalente al automóvil para el transporte humano, mientras que las grandes estaciones terrenas son como autobuses o trenes.

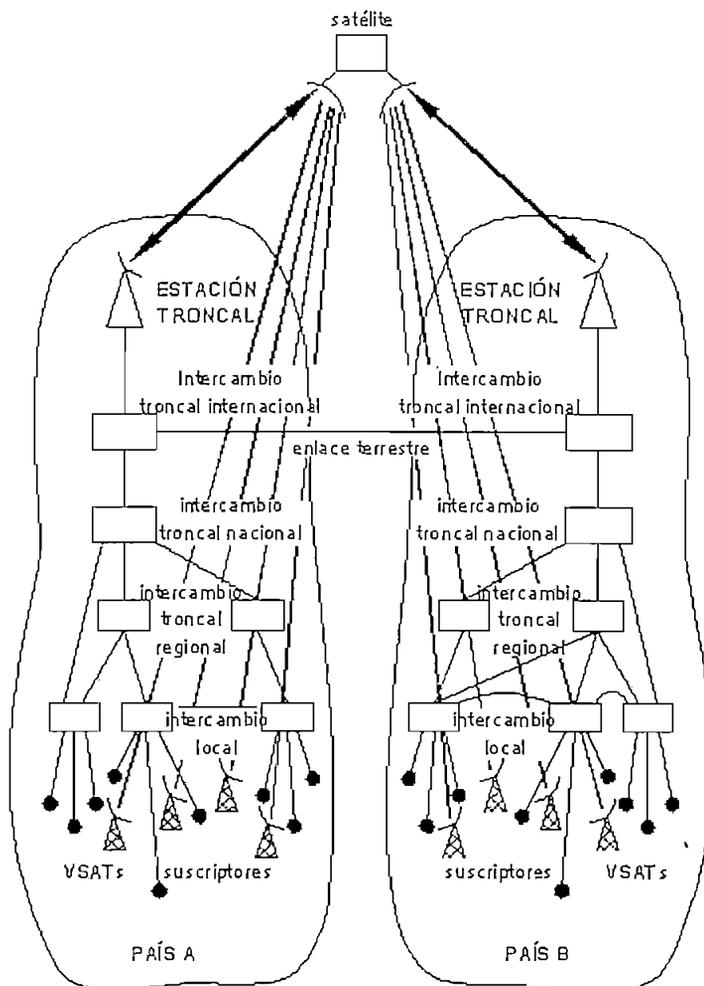


Figura 1.3 Desde estaciones troncales hasta VSATs.

En este punto vale la pena observar que las VSAT, así como los automóviles, están a completa disposición de su dueño, esto evita la necesidad de usar un enlace a través de la red pública para acceder a la estación terrena. De hecho, el usuario puede conectar directamente en su equipo VSAT, sus propios terminales de comunicación tal como un teléfono, un computador, una impresora, un equipo de vídeo, etc.

Por consiguiente, las comunicaciones VSAT aparecen como medios naturales para eludir a los operadores de la red pública, accediendo directamente al satélite. Este procedimiento es conocido en nuestro medio como "bypass".

Éstas son herramientas flexibles para establecer redes privadas, por ejemplo entre los diferentes sitios de una compañía. La figura 1.3, ilustra este aspecto destacando el posicionamiento de las VSAT cerca del usuario comparado con las estaciones troncales, las cuales se localizan al nivel más alto de la jerarquía de la red pública conmutada.

La oportunidad de *bypass* ofrecida por las redes VSAT no siempre ha sido bien acogida por los operadores nacionales, ya que ésta podría significar la pérdida de ingresos, como resultado del tráfico desviado de la red pública. Esto ha generado políticas conservadoras por parte de los operadores de telecomunicaciones nacionales, que se oponen a la desreglamentación del sector de las comunicaciones. En algunas regiones del mundo, y particularmente en Europa, esto ha sido una fuerte restricción al desarrollo de las redes VSAT.

1.1.1 CONFIGURACIONES DE LAS REDES VSAT

Como se muestra en la figura 1.4, las VSAT son conectadas por enlaces de radiofrecuencia vía satélite llamados *uplink* desde la estación al satélite y *downlink* desde el satélite a la estación. El enlace global de una estación a otra, llamado *hop* (salto), consiste de un *uplink* y un *downlink*.

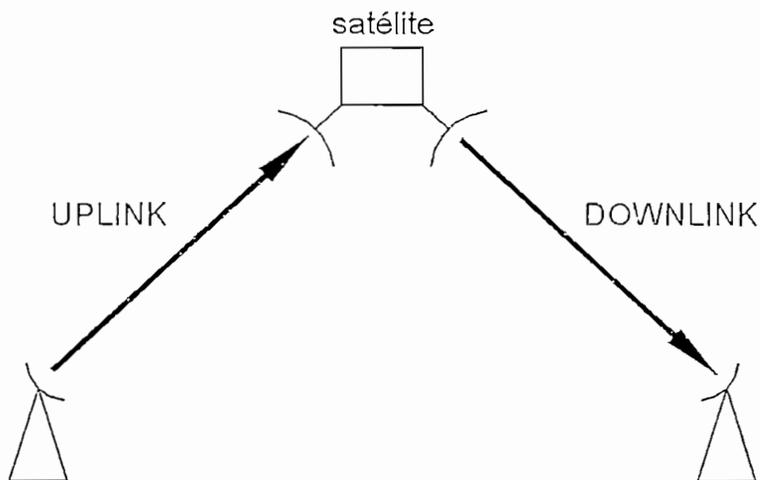


Figura 1.4 Definición de *uplink* y *downlink*.

El satélite básicamente, recibe las portadoras del enlace *uplink* desde las estaciones terrenas transmisoras que están dentro del campo de vista de su antena receptora, las amplifica, cambia su frecuencia a una banda más baja para evitar una posible interferencia de entrada/salida, y las transmite amplificadas a las estaciones localizadas dentro del campo de vista de su antena transmisora.

Las redes VSAT usan satélites geostacionarios, que son aquellos que orbitan en el plano ecuatorial a una altitud sobre la superficie de la tierra de 35786 metros. Como el satélite se mueve en su órbita en la misma dirección que el giro de la tierra, éste aparece para cualquier estación terrena como un punto fijo en el cielo. La figura 1.5 ilustra esta geometría.

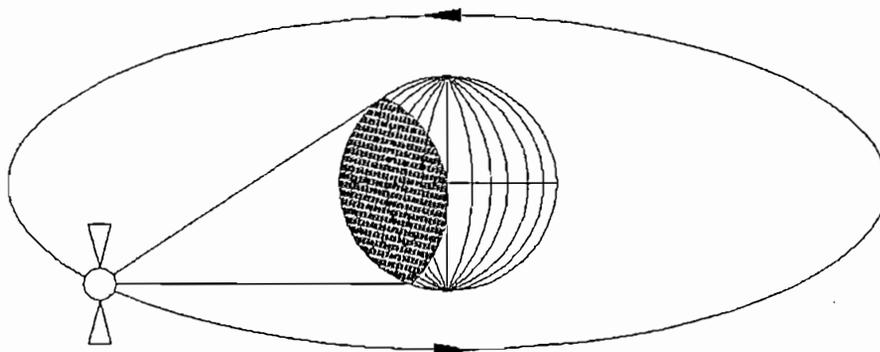


Figura 1.5 Satélite Geoestacionario.

Debe notarse que la distancia de una estación terrena al satélite geoestacionario induce una atenuación en la potencia de la portadora de radiofrecuencia, típicamente de 200 dB en los enlaces, *uplink* y *downlink*; y, un retardo de propagación entre estaciones terrenas (*hop delay*) de aproximadamente 0.25 segundos.

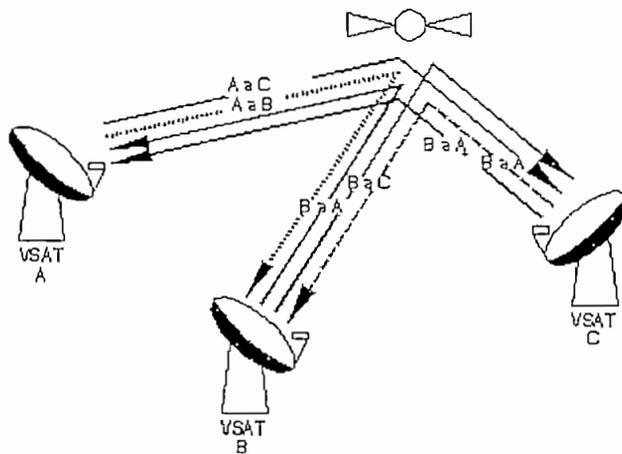
Como consecuencia de su aparente posición fija en el cielo, el satélite puede usarse las 24 horas del día como un *relay* permanente de portadoras de radiofrecuencia del enlace *uplink*. Esas portadoras son enviadas por el enlace *downlink* a todas las estaciones terrenas visibles desde el satélite (el área sombreada en la Tierra en la figura 1.5).

1.1.1.1 Configuración en malla

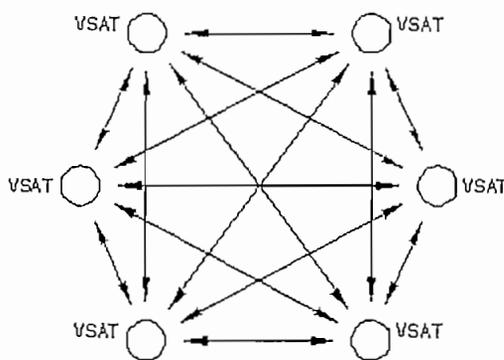
En este caso las portadoras de una VSAT pueden ser relevadas a cualquier otra estación dentro de la red, ya que todas son visibles desde el satélite, tal como se indica en la figura 1.6, la cual hace referencia a una configuración en malla de una red VSAT.

Con respecto a esta configuración se deben tener en cuenta las siguientes limitaciones:

- Atenuación de la potencia de la portadora típicamente de 200 dB en los enlaces de *uplink* y *downlink*, como resultado de la distancia al satélite geoestacionario;
- Potencia limitada a unas pocas decenas de vatios del *transponder* satelital de radiofrecuencia;
- Tamaño reducido de las VSAT, lo que limita su potencia de transmisión y su sensibilidad de recepción.



(a)



(b)

Figura 1.6 Configuración en malla.

Como resultado de lo anterior, puede ser que las señales demoduladas en el receptor de la VSAT no alcancen la calidad requerida por los terminales de usuarios. Por consiguiente los enlaces directos entre estaciones VSAT no son mayormente utilizados.

1.1.1.2 Configuración en estrella

La solución a las limitaciones de la configuración en malla, es entonces instalar en la red una estación más grande que una VSAT, llamada *hub*.

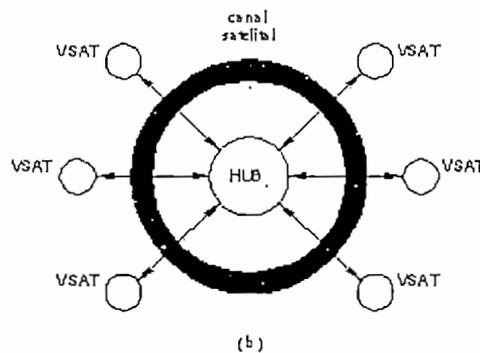
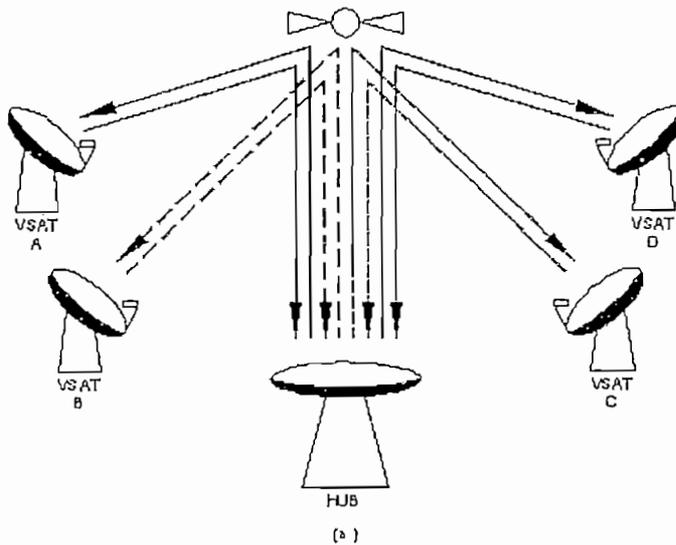


Figura 1.7 Configuración en estrella bidireccional.

La estación *hub* tiene una antena más grande que la de una VSAT, generalmente de 4 a 11 metros, resultando esto en una mayor ganancia; además, está provista con un transmisor más poderoso.

Como resultado de su capacidad mejorada, la estación *hub* puede recibir adecuadamente todas las portadoras transmitidas por las VSAT, y transmitir la información deseada a todas las VSAT por medio de sus propias portadoras. De esta manera, la arquitectura de la red se convierte en una estrella como se muestra en las figuras 1.7 y 1.8.

Los enlaces desde el *hub* hacia la estación VSAT son llamados *outbound links* (enlaces salientes), y desde la VSAT hacia el *hub*, *inbound links* (enlaces

entrantes). Cada uno de ellos consiste a su vez, de dos enlaces, uno *uplink* y otro *downlink*, hacia el satélite y desde él, respectivamente.

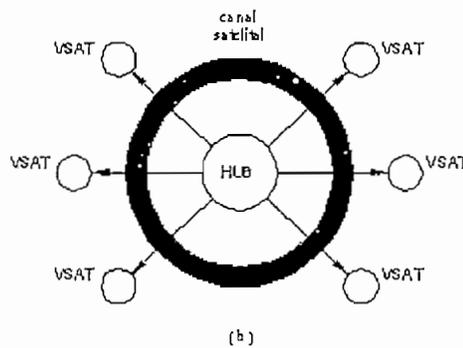
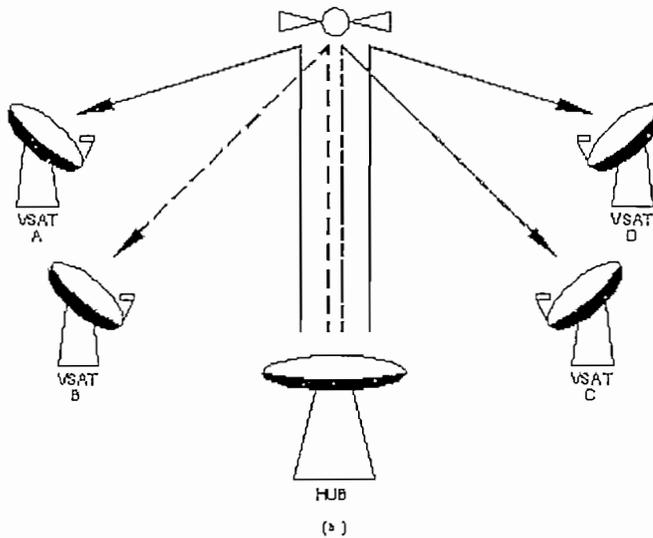


Figura 1.8 Configuración en estrella unidireccional.

Existen dos tipos de red VSAT en estrella:

- *Redes bidireccionales* (figura 1.7). Son aquellas donde las VSAT pueden transmitir y recibir. Tales redes soportan tráfico interactivo.
- *Redes unidireccionales* (figura 1.8). Son aquellas donde las VSAT son estaciones de solo recepción, y hacia las cuales el *hub*, transmite sus portadoras. Esta configuración soporta servicios de radiodifusión, desde un

sitio central en el cual el *hub* está localizado, hacia sitios remotos donde las estaciones VSAT son instaladas.

1.1.1.3 Comparación entre configuraciones malla y estrella

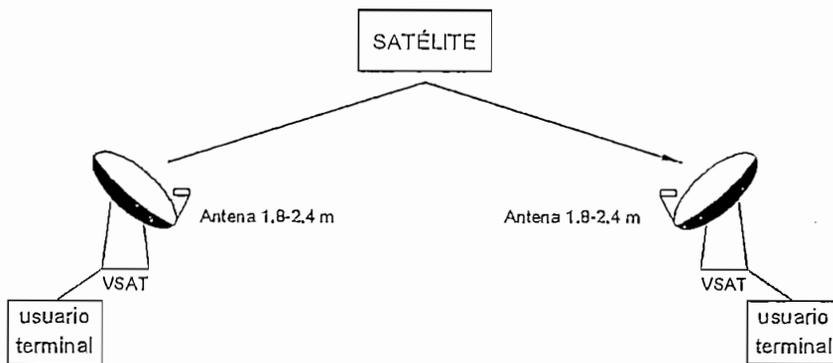


Figura 1.9 Conectividad del usuario terminal dentro de una red VSAT malla.

Las figuras 1.9 y 1.10, indican una antena más pequeña para las estaciones VSAT dentro de una configuración en estrella que para aquellas dentro de una red en malla.

Esto se debe a la inclusión del *hub* en la configuración en estrella, que proporciona mayor potencia en el *outbound link* y una mejor recepción de las portadoras transmitidas por las VSAT en el *inbound link*, que en una configuración en malla. Es por eso que, la antena del *hub* es mucho más grande que las antenas de las estaciones terrenas.

En conclusión, las redes en estrella son impuestas por las limitaciones de potencia resultado del tamaño reducido de las estaciones terrenas VSAT, en conjunto con la limitación de potencia del *transponder* del satélite. Esto es particularmente verdadero cuando se desea una baja inversión económica al adquirir las estaciones VSAT. Se consideran las redes en malla siempre que el factor económico no represente una limitación.

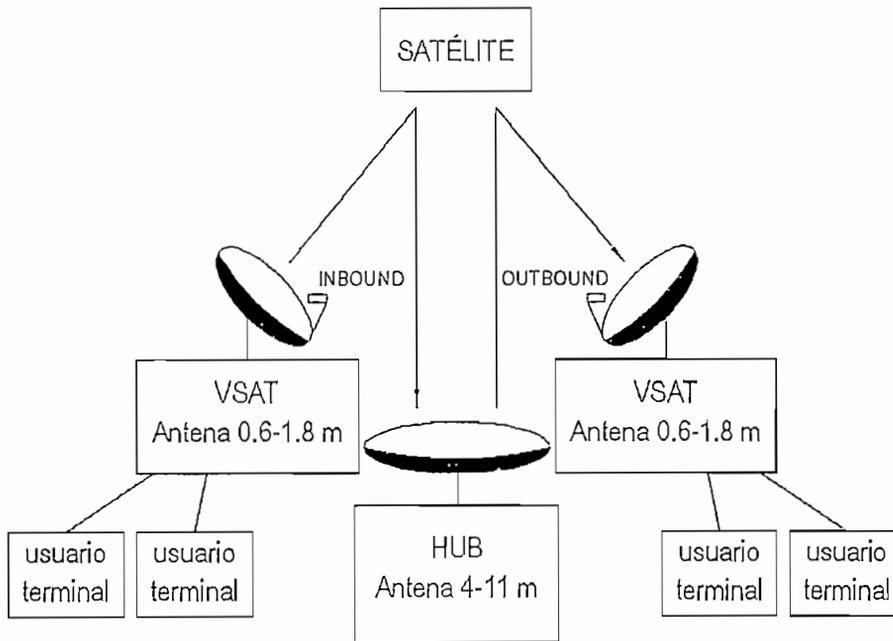


Figura 1.10 Conectividad del usuario terminal usando el *hub* como un relevador en una red VSAT estrella.

Las redes en malla tienen la ventaja de un retardo de propagación reducido (0.25 s en un solo salto en lugar de 0.5 s para un doble salto), que es de interés, sobre todo para los servicios de telefonía.

1.1.2 ESTACIONES TERRENAS DE LAS REDES VSAT

1.1.2.1 Estación VSAT

La figura 1.11, muestra la arquitectura de una estación VSAT. Como se puede observar, una estación VSAT está compuesta de dos conjuntos de equipos separados: la *unidad externa* (ODU *OutDoor Unit*) y la *unidad interna* (IDU *InDoor Unit*).

La ODU es el interfaz de la VSAT al satélite, mientras que la IDU es el interfaz a los terminales del cliente o a la red de área local (LAN).

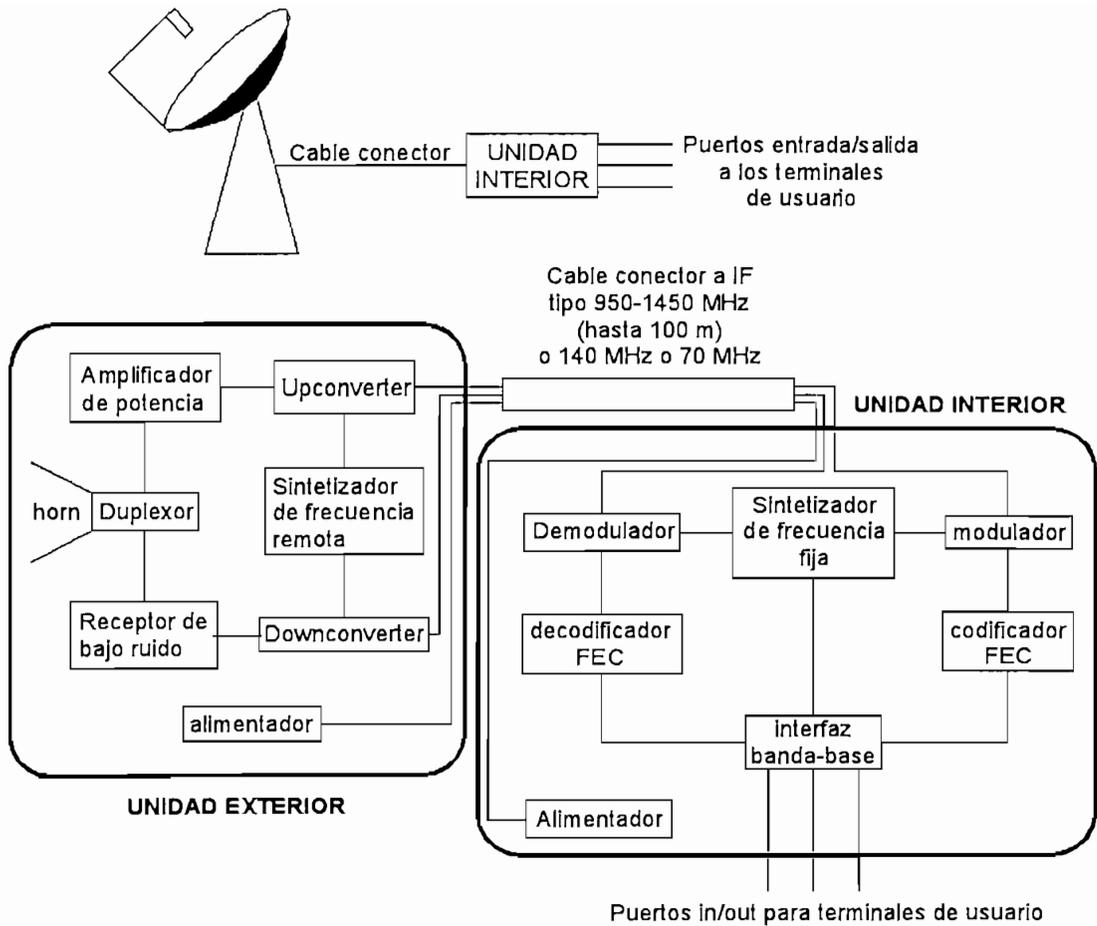


Figura 1.11 Diagrama de bloques de una estación terrena. [2]

1.1.2.1.1 La unidad externa (ODU)

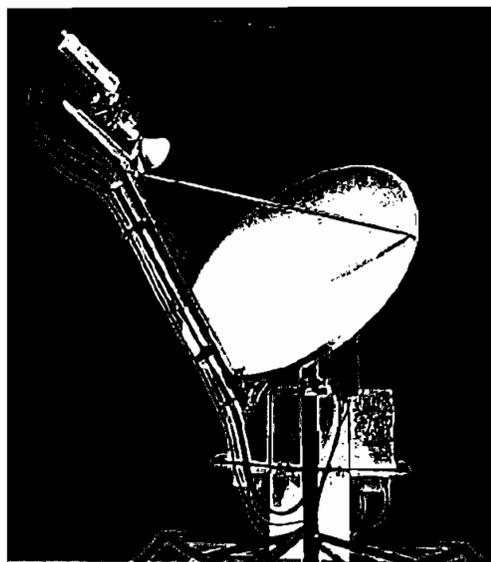


Figura 1.12 Fotografía de una ODU de una estación VSAT. [3]

La figura 1.12, muestra la fotografía de la unidad externa de una VSAT, con su antena y el contenedor electrónico que empaqueta el amplificador transmisor, el receptor de bajo ruido, los convertidores de subida y bajada, y el sintetizador de frecuencia. La fotografía de la figura 1.13, proporciona una vista más cercana de este contenedor.

Para una apropiada especificación de la ODU como un interfaz al satélite, se deben tomar en cuenta los siguientes parámetros:

- Las bandas de frecuencia de transmisión y recepción.
- La resolución o tamaño de paso del transmisor y receptor para ajustar la frecuencia de la portadora transmitida o para sintonizar la frecuencia de la portadora recibida.
- La potencia isotrópica radiada equivalente (PIRE) que determina el desempeño del enlace de radiofrecuencia de la estación hacia el satélite. La PIRE depende del valor de la ganancia de la antena, de su tamaño, frecuencia de transmisión, y potencia de salida del amplificador de transmisión.
- La relación G/T [dB/K] que determina el desempeño del *uplink*. Ésta depende del valor de la ganancia de la antena, de su tamaño, frecuencia de recepción, y de la temperatura de ruido del receptor.
- La variación de la ganancia de los lóbulos de la antena al ponerla fuera del eje que la direcciona hacia el satélite, con la cual se controla la PIRE y G/T bajo estas condiciones, determinando así, los niveles de interferencia producidos y recibidos.

También serán considerados parámetros como el rango de temperatura de funcionamiento, lluvia, humedad, etc.

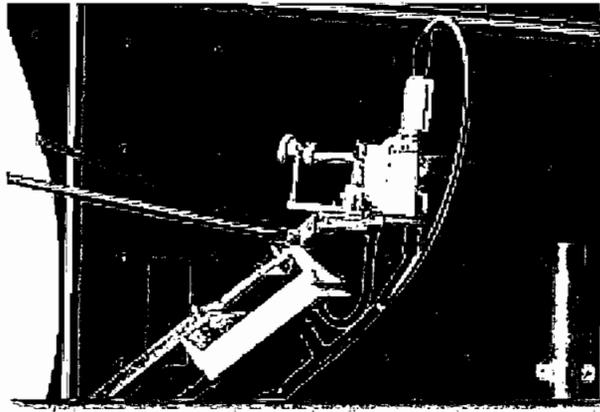


Figura 1.13 Fotografía de un contenedor electrónico. [4]

La tabla 1.1 indica los valores típicos de los parámetros de una ODU.

Banda de frecuencia de transmisión	14.0 – 14.5 GHz (Ku-band)
	5.925 – 6.425 GHz (C-band)
Banda de frecuencia de recepción	10.7 – 12.75 GHz (Ku-band)
	3.625 – 4.2 GHz (C-band)
<i>Antena</i>	
Tipo de antena	<i>Offset</i> , reflector simple, montada fija
Diámetro	1.8 – 3.5 m en la C-band
	1.2 – 1.8 m en la Ku-band
Aislamiento Tx/Rx	35 dB
Voltaje <i>Standing Wave</i>	1.3:1
<i>Radio (VSWR)</i>	
Polarización	Ortogonal lineal en la Ku-band
	Ortogonal circular en la C-band
Ajuste de polarización	$\pm 90^\circ$ para polarización lineal de la antena
Aislamiento de polarización cruzada	30 dB en el eje, 22 dB dentro de 1 dB beamwidth
	17 dB de 1° a 10° fuera del eje
Envoltura del lado del lóbulo	$29 - 25 \log \theta$
Ajuste de azimut	160 grados continuos, con ajuste fino
Ángulo de elevación	3 a 90 grados
Posicionamiento	Posicionamiento automático opcional
<i>Tracking</i>	Ninguno
<i>Velocidad del viento</i>	
Operación	75 a 100 km/h

Supervivencia	210 km/h
<i>Deicer</i>	Eléctrico (opcional) o pasivo (abrigo hydrofóbico)
<i>Amplificador de potencia</i>	
Potencia de salida	0,5 W a 5 W SSPA en la Ku-band
	3 W a 30 W en la C-band
Pasos de frecuencia	100 kHz
<i>Receptor de bajo ruido</i>	
Temperatura de ruido	80 – 120 K en la Ku-band
	35 – 55 K en la C-band
<i>Características generales</i>	
Potencia Efectiva Isotrópica Radiada	44 a 55 dBW en la C-band
	43 a 53 dBW en la Ku-band
Figura de mérito G/T	13 a 14 dBK ⁻¹ en la C-band
	19 a 23 dBK ⁻¹ en la Ku-band (cielo despejado)
	14 a 18 dBK ⁻¹ en la Ku-band (99.99% del tiempo)
Temperatura de operación	-30°C a +55°C

Tabla 1.1 Valores típicos para las partes de la ODU de una estación VSAT.

1.1.2.1.2 La unidad interna (IDU)

La unidad interna instalada en la localidad del usuario, se muestra en la figura 1.14. Para conectar sus terminales a la VSAT, el usuario debe acceder a los puertos instalados en el tablero posterior de la IDU.

Para una especificación apropiada de la IDU, como una interfaz a los terminales de usuario o a una red de área local (LAN), se deben considerar los siguientes parámetros:

- Número de puertos.
- Tipo de puertos: interfaz mecánica, eléctrica, funcional y de procedimiento.

- Velocidad del puerto: ésta es la velocidad máxima a la que pueden intercambiarse los datos entre el terminal del usuario y la IDU en un puerto dado. La velocidad de transmisión real puede ser más baja.



Figura 1.14 Vista frontal de una IDU de una estación VSAT. [5]

Se usa un esquema de modulación coherente tal como el BPSK (*BiPhase Shift Keying*) o el QPSK (*Quadrature Phase Shift Keying*). Para un adecuado desempeño, la velocidad de transmisión de la portadora debe ser superior a los 2,4 Kbps, de otra forma el ruido de fase se vuelve un problema.

Para velocidades de transmisión más bajas, se evita usar una modulación de fase y en cambio se usa una modulación de frecuencia FSK (*Frequency Shift Keying*).

1.1.2.2 Estación *Hub*

La figura 1.15 muestra una fotografía de una estación *hub* y la figura 1.16 muestra la arquitectura de la estación *hub* con su equipamiento. Aparte del tamaño y el número de subsistemas, la diferencia funcional entre un *hub* y una VSAT es mínima.

La mayor diferencia entre una estación VSAT y una *hub* es que los interfaces de la IDU de esta última a una computadora central o, a una red pública conmutada o privada, por ejemplo, varían dependiendo si éstos son dedicados o compartidos. En la tabla 1.2 se indican los parámetros de la ODU de una estación *hub*.

Banda de frecuencia de transmisión	14.0 – 14.5 GHz (Ku-band)
	5.925 – 6.425 GHz (C-band)
Banda de frecuencia de recepción	10.7 – 12.75 GHz (Ku-band)
	3.625 – 4.2 GHz (C-band)
<i>Antena</i>	
Tipo de antena	Reflector dual asimétrico (<i>Cassegain</i>)
Diámetro	2 – 5 m (<i>hub</i> compacto)
	5 – 8 m (<i>hub</i> mediano)
	8 – 10 m (<i>hub</i> grande)
Aislamiento Tx/Rx	30 dB
Voltaje <i>Standing Wave</i>	1.25:1
<i>Radio (VSWR)</i>	
Polarización	Ortogonal lineal en la Ku-band
	Ortogonal circular en la C-band
Ajuste de polarización	$\pm 90^\circ$ para polarización lineal de la antena
Aislamiento de polarización cruzada	30 dB en el eje
Envoltura del lado del lóbulo	$29 - 25 \log \theta$
Ángulo de azimut	120 grados
Ángulo de elevación	3 a 90 grados
Posicionamiento	0.01 °/s
<i>Tracking</i>	<i>Steptrack</i> en la Ku-band si la antena es mayor a 4 m
<i>Velocidad del viento</i>	
Operación	50 a 70 km/h
Supervivencia	180 km/h
<i>Deicer</i>	Eléctrico
<i>Amplificador de potencia</i>	
Potencia de salida	3 W a 15 W SSPA en la Ku-band
	5 W a 20 W SSPA en la C-band
	50 W a 100 W TWT en la Ku-band
	100W a 200 W TWT en la C-band
Pasos de frecuencia	100 kHz a 500 kHz
<i>Receptor de bajo ruido</i>	
Temperatura de ruido	80 – 120 K en la Ku-band
	35 – 55 K en la C-band
Temperatura de operación	-30°C a +55°C

Tabla 1.2. Valores típicos para las partes de la ODU de una estación *hub*.



Figura 1.15 Fotografía de una estación *hub*. [1]

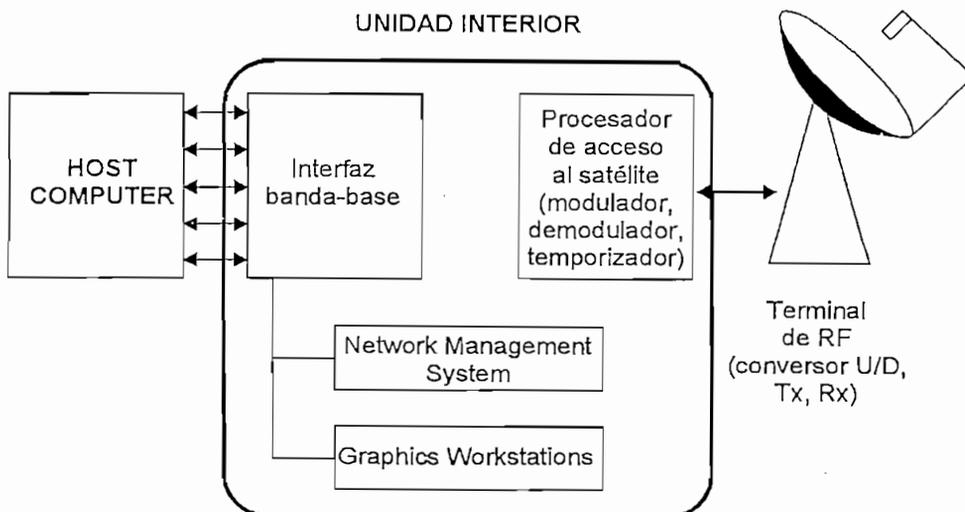


Figura 1.16 Subsistemas del *hub*.

Se puede notar en la figura 1.16 que la estación *hub* está provista de un sistema de administración de red denominado NMS (*Network Management System*).

El NMS es un mini computador o una estación de trabajo equipada con un *software* especializado, usado para realizar funciones operacionales y administrativas. Este mini computador está conectado a cada VSAT en la red por medio de circuitos virtuales permanentes. Constantemente se intercambian mensajes de administración entre el NMS y las VSAT, y contienen con el tráfico normal para acceder a los recursos de la red.

1.1.2.2.1 *Funciones operacionales*

Las funciones operacionales están relacionadas al manejo de la red y proporcionan la capacidad para reconfigurarla dinámicamente, agregando o eliminando estaciones VSAT, portadoras e interfaces de red.

Las funciones operacionales también incluyen una supervisión y control del desempeño y estado del *hub*, de cada estación VSAT y de todos los puertos de datos asociados a la red. Esto conlleva a las funciones administrativas, las cuales proveen la asignación en tiempo real y conectividad de las VSAT y, el manejo y control de nuevas instalaciones y configuraciones.

El *software* de control de red permite la asignación dinámica y automática de capacidad a las VSAT que trabajan con tráfico interactivo a ráfagas y a las que ocasionalmente se usarán para trabajar con tráfico continuo. Ninguna intervención del operador se requiere para efectuar esta reasignación de capacidad temporal.

El NMS notifica al operador en caso de saturación de la capacidad, lo que impediría a más usuarios VSAT acceder al servicio. El NMS también maneja todos los aspectos relacionados con alarmas y diagnóstico de fallas. En particular, en caso de cualquier interrupción de energía en las estaciones VSAT, el NMS

recupera todo el *software* y parámetros necesarios del sistema para un reinicio de operación.

1.1.2.2.2 *Funciones administrativas*

Las funciones administrativas se encargan del inventario de equipo, archivos de uso de la red, seguridad y facturación.

El NMS mantiene un registro de las estaciones VSAT instaladas y aquellas que se encuentran en operación, la configuración del equipo dentro del *hub* y de cada estación VSAT, así como la configuración de los puertos de cada interfaz de red.

Esta información está disponible para el operador, junto con la información estadística de tráfico, número de fallas, tiempo promedio de retardo en la transmisión de datos, etc. La información puede analizarse e imprimirse diaria, semanal o mensualmente; así como puede ser guardada magnéticamente para referencias futuras. Las funciones administrativas forman la base para el análisis de desempeño de tráfico y tendencia, distribución de costos basados en uso, etc.

La larga y diversa lista anterior de funciones a realizar por el NMS muestra su importancia para la red. Realmente, la suficiencia de respuesta del NMS a las necesidades de los usuarios marca la diferencia entre los proveedores de red reconocidos, y aquellos que no sobreviven en el mercado.

1.1.3 APLICACIONES DE LAS REDES VSAT Y TIPOS DE TRÁFICO

Dependiendo de las necesidades del cliente, éste transmitirá ya sea un tipo de señal o una mezcla de ellas.

Se transmiten datos y voz en un formato digital, mientras que el vídeo puede ser analógico o digital. Cuando es digital, la señal de vídeo se beneficia de las técnicas de compresión de ancho de banda.

1.1.3.1 Comunicaciones de datos

Las VSAT han surgido de la necesidad de transmitir datos. Los productos estándares VSAT ofrecen los medios para la transmisión de los mismos.

Las velocidades ofrecidas a un usuario están dentro del rango de los 50 bps a los 64 kbps, para velocidades menores a los 20 kbps se usan puertos con interfaz RS-232, y para velocidades superiores los interfaces RS-422, V.35 y X.21.

La distribución de datos puede ser implementada en combinación con la transmisión de vídeo, usando el estándar DVB-S (*Digital Video Broadcasting by Satellite*) por ejemplo.

1.1.3.2 Comunicaciones de voz

Las comunicaciones de voz son de interés en las redes bidireccionales solamente. Se las puede implementar a velocidades bajas usando la codificación de voz (*vocoders*). La velocidad de información está en el rango de los 4,8 kbps a los 9,6 kbps. También pueden combinarse con las comunicaciones de datos; por ejemplo, se pueden multiplexar hasta 4 canales de voz con datos o con canales facsímiles en un solo canal de 64 kbps.

En las redes VSAT las comunicaciones de voz sufren un retardo asociado al procesamiento del *vocoder* (aproximadamente 50 ms) y a la propagación en los enlaces satelitales (aproximadamente 500 ms para un salto doble). Por consiguiente, el usuario puede preferir conectarse a redes terrestres que ofrecen un retardo reducido.

Las comunicaciones de voz pueden ser una oportunidad en el mercado para las VSAT como un servicio brindado a localidades donde las líneas terrestres no están disponibles, o para aplicaciones terminales móviles.

1.1.3.3 Comunicaciones de vídeo

1.1.3.3.1 *En el enlace outbound*

Las comunicaciones de vídeo hacen uso de los estándares más comunes de televisión (NTSC, PAL o SECAM) en combinación con una modulación FM, o pueden ser implementadas usando el estándar DVB-S, posiblemente en combinación con la distribución de datos.

1.1.3.3.2 *En el enlace inbound*

Como resultado de la limitada potencia de las VSAT en el *uplink*, la transmisión de vídeo solo es posible a bajas velocidades. Entonces, ésta será probablemente en forma de cámara lenta, usando codificación y compresión de vídeo.

Desde otro punto de vista, las redes VSAT tienen dos aplicaciones: civiles y militares.

1.1.3.4 Redes VSAT para aplicaciones civiles

1.1.3.4.1 *Tipos de servicio*

Como se mencionó anteriormente, las redes VSAT pueden ser configuradas como redes unidireccionales o bidireccionales. La siguiente lista muestra ejemplos de servicios soportados por redes VSAT según estos dos tipos de configuración.

Redes VSAT unidireccionales

- Disponibilidad de mercado y otras noticias de difusión
- Entrenamiento o educación continua a distancia
- Distribución de tendencias financieras y documentos

- Presentación de nuevos productos en localidades geográficamente dispersas
- Distribución de vídeo o programas de TV
- Distribución de música disponible y comerciales de una tienda

Redes VSAT bidireccionales

- Transacciones computacionales interactivas
- Videoconferencia de baja velocidad
- Acceso a bases de datos
- Transacciones bancarias, máquinas contestadoras automáticas, puntos de venta
- Sistemas de reservación
- Monitoreo de ventas, control de inventarios
- Control de procesos remotos distribuidos y telemetría
- Datos médicos, transferencia de imágenes
- Recolección de noticias a través del satélite
- Videoteleconferencia

Puede notarse que la mayoría de los servicios soportados por redes VSAT bidireccionales trabajan con un tráfico de datos interactivo, donde los terminales de usuario son en su mayoría computadores personales. Las excepciones más notables son comunicaciones de voz y recolección de noticias a través de satélite.

Las comunicaciones de voz en una red VSAT implican una telefonía con mayores retardos que aquellos presentes en las líneas terrestres, como resultado de la gran distancia hacia el satélite. Los servicios de telefonía involucran una conectividad completa, y los retardos son típicamente de 0.25 s ó 0.50 s dependiendo de la configuración de la red seleccionada.

La recolección de noticias a través del satélite puede verse como una red temporal que usa estaciones VSAT móviles, a veces llamadas estaciones *fly-*

away, que se transportan por automóvil o avión, y se instalan en un lugar donde los reporteros de noticias pueden transmitir las señales de vídeo al *hub* localizado cerca del estudio de la compañía.

De hecho, el servicio podría ser considerado únicamente como entrante (*inbound*), si no fuera por la necesidad de verificar el *uplink* del sitio remoto y estar en contacto por teléfono con el personal del estudio. Por ejemplo, el reportero solicitará al estudio central le confirmen vía telefónica si se ha establecido la conexión para iniciar la transmisión.

Como las estaciones *fly-away* son constantemente transportadas, montadas y desmontadas, deben ser robustas, livianas y fáciles de instalar. Hoy en día éstas pesan típicamente 100 kg y pueden instalarse en menos de 20 minutos.

1.1.3.4.2 Tipos de tráfico

Dependiendo del servicio, el flujo de tráfico entre el *hub* y las VSAT puede tener diferentes características y requerimientos, tal y como se muestra a continuación:

Tráfico de transferencia de datos o broadcasting, que pertenece a la categoría de servicios unidireccionales, típicamente trabaja con transferencia de archivos de datos que van de uno a cien megabytes. Este tipo de servicio no es sensible al retardo, pero requiere de una alta integridad de los datos que son transferidos, por ejemplo, descargas de archivos de computadores y distribución de datos a sitios remotos.

Tráfico de datos interactivos, servicio bidireccional que corresponde a varias transacciones por minuto y por terminal de simples paquetes que van de 50 a 250 *bytes* de longitud en ambos enlaces, *inbound* y *outbound*. El tiempo de respuesta esperado es, por lo general, de unos pocos segundos, por ejemplo, transacciones bancarias y transferencia electrónica de fondos en puntos de venta.

Tráfico de pregunta/respuesta, servicio bidireccional que corresponde a varias transacciones por minuto y terminal. Los paquetes en el *inbound* (usualmente de 30 a 100 *bytes*) son más cortos que los paquetes en el *outbound* (usualmente de 500 a 2000 *bytes*). El tiempo de respuesta esperado es de unos pocos segundos, por ejemplo, reservaciones en aerolíneas u hoteles y consultas a bases de datos.

Tráfico de control de supervisión y adquisición de datos (SCADA), servicio bidireccional correspondiente a una transacción por segundo o minuto por terminal. Los paquetes en el *inbound* (generalmente de 100 *bytes*) son más grandes que los paquetes en el *outbound* (generalmente de 10 *bytes*). El rango de tiempo de respuesta esperado va de unos pocos segundos a pocos minutos. Algo que es muy importante es el alto nivel de seguridad de datos y el bajo consumo de potencia del terminal, por ejemplo, aplicaciones de control y monitoreo de tuberías, dispositivos eléctricos y recursos hidráulicos.

1.1.3.5 Redes VSAT para aplicaciones militares

Las redes VSAT han sido adoptadas por muchas fuerzas militares en el mundo. De hecho, la flexibilidad inherente con el despliegue en el desarrollo de las VSAT, ha hecho de ellas un valioso medio para instalar enlaces de comunicaciones temporales entre unidades pequeñas en el campo de batalla y los cuarteles localizados cerca del *hub*. Es más, la configuración de red en estrella se adapta al tipo de flujo de información existente entre las unidades de campo y la base de comandos.

Las bandas de frecuencia pertenecen a la banda X, con *uplinks* en la banda de los 7.9 - 8.4 GHz y *downlinks* en la banda de los 7.25 - 7.75 GHz.

Las estaciones VSAT para uso militar deben ser pequeñas, livianas y de bajo consumo de potencia, lo cual facilita su operación bajo condiciones de batalla. Como ejemplo, la estación *Manpack* desarrollada por la *UK Defence*

Research Agency (DRA) para su red militar VSAT llamada *Milpico*, está equipada con una antena de 45 centímetros; su peso es menor a los 17 kg y puede instalarse en 90 segundos. Esta estación soporta datos y voz codificada a una velocidad de 2.4 kbps.

Para lograr lo antes mencionado, las estaciones *hub* necesitan ser equipadas con antenas tan grandes como 14 metros. Otro requisito importante es tener una baja probabilidad de detección por interceptores hostiles. Las técnicas de espectro expandido son muy usadas en este tipo de aplicaciones.

1.1.4 VENTAJAS Y DESVENTAJAS DE UNA RED VSAT

1.1.4.1 Ventajas

- Comunicaciones punto a multipunto y punto a punto. Las redes VSAT ofrecen la posibilidad de configurarlas de dos maneras: en estrella (punto a multipunto) y en malla (punto a punto), tal como se mencionó anteriormente.
- Asimetría de transferencia de datos. Las redes VSAT ofrecen diferentes capacidades tanto en el enlace *inbound* como en el *outbound*. Esto representa una ventaja considerando la naturaleza asimétrica de las aplicaciones de usuario. Por ejemplo, si se ofreciera igual capacidad en ambos enlaces, el cliente tendría que pagar por una capacidad no utilizada.
- Flexibilidad. Las redes VSAT, por su naturaleza, proveen facilidades para reconfiguración y crecimiento (adición y remoción de estaciones), resultado de su fácil desplazamiento e instalación.
- Redes corporativas privadas. Una red VSAT ofrece a su operador un control extremo a extremo sobre la calidad de la transmisión y la confiabilidad del enlace. También lo protege de posibles e inesperadas fluctuaciones de aranceles, ofreciendo estabilidad de precios y la posibilidad de prever sus

gastos de comunicación. Por consiguiente es un apoyo adecuado a las redes corporativas privadas.

- Baja Tasa de Bits Errados (BER). Las redes VSAT tienen un excelente BER generalmente de 10^{-7} .
- Costo independiente de la distancia. El costo de un enlace en una red VSAT no depende de las distancias. De ahí que, se espera un ahorro de costos si la red muestra un gran número de estaciones remotas y una dispersión geográfica alta.

1.1.4.2 Desventajas

- Sensibilidad a la interferencia. Un enlace de radiofrecuencia en una red VSAT está sujeto a la interferencia como resultado del reducido tamaño de las antenas en las estaciones terrenas.
- Intrusos. Como consecuencia de la gran cobertura de un satélite geostacionario, puede ser fácil para un intruso acceder a la información que éste difunde, demodulando la portadora. Por consiguiente, para prevenir el uso desautorizado de la información transportada en la portadora satelital, obligatoriamente, ésta debe ser encriptada.
- La pérdida de un *transponder* puede llevar a la pérdida de la red. El satélite es considerado como un punto simple de falla, entonces, si su *transponder* se avería, la red VSAT dejaría de funcionar. Los enlaces de comunicación pueden ser restaurados usando un *transponder* de respaldo, en el mismo satélite; así, con un pequeño cambio en la frecuencia o en la polarización se devolvería rápidamente el funcionamiento a la red. Sin embargo, ese *transponder* debería encontrarse en otro satélite, pudiendo significar esto un reajuste de la posición de la antena de cada una de las estaciones terrenas, y esto puede tardar algún tiempo.

- Retardo de propagación (doble salto = 0.5 s). El tiempo de propagación entre estaciones terrenas dentro de una red en estrella, impone un doble salto, lo cual implica un retardo de medio segundo en la transferencia de información. Esto limita el uso de comunicaciones de voz, por ejemplo, en las redes VSAT.

1.2 REDES SATELITALES DE BANDA ANCHA

La velocidad con la cual se puede transportar información sobre una conexión de telecomunicaciones está limitada por su ancho de banda. Entre mayor es éste, mayor será la velocidad máxima de transmisión de la información. Pero además, los diferentes tipos de servicios que se pueden manejar estarán influenciados por la manera de operar de la red.

Una red de banda ancha proporciona una amplia gama de velocidades y gran variedad de diferentes exigencias de servicio. En esencia, una red de banda ancha es aquella que soporta servicios con velocidades que superan la velocidad primaria de 64 Kbps (nx64 Kbps). [1]

La tecnología que apoya el servicio de banda ancha está en constante desarrollo; su objetivo es abarcar múltiples servicios, también llamados multiservicios, que incluyen vídeo, televisión y datos de alta velocidad, particularmente cuando se necesitan imágenes de muy alta resolución o tiempos rápidos de respuesta.

Existen tecnologías ya desarrolladas, capaces de proporcionar gran ancho de banda para soportar los multiservicios. Entre éstas se tienen: *TDM (Time Division Multiplexing)*, *Frame Relay*, *RDSI* o *ISDN (Integrated Services Digital Network)*, redes de fibra óptica, *ADSL (Asymmetric Digital Subscriber Line)* y *SDSL (Symmetric Digital Subscriber Line)*, accesorios inalámbricos de alta velocidad, entre otras.

Los conceptos anteriores son aplicables también a las comunicaciones satelitales, es decir que, bajo las condiciones mencionadas, se puede hablar de redes satelitales de banda ancha, las cuales buscan soportar la misma variedad de servicios que sus homónimas terrestres.

El principal limitante al que se enfrentan las redes satelitales de banda ancha es la gran complejidad que implica soportar la característica de multiservicios, especialmente aquellos en tiempo real, tales como videoconferencia o VoIP (voz sobre IP).

Debido a este problema, las redes satelitales, en su mayoría, brindan solo servicios caracterizados por tráfico asimétrico, tal como acceso a *Internet*; servicios de *pay-per-view*, compras desde el hogar, comercio electrónico, entre otros. Por tanto estas redes dejan a un lado aquellos servicios de tiempo real que se caracterizan por un tráfico simétrico. [6]

El esquema utilizado para este fin se muestra en la figura 1.17. Como se puede observar, este esquema consta de un canal satelital por el cual se difunde los servicios requeridos por los usuarios desde el proveedor hacia los mismos, usando las normativas contempladas en el estándar DVB-S; y un canal de retorno terrestre utilizado por el usuario para interactuar con el proveedor de servicios a través de peticiones y solicitudes.

Este esquema se justifica en cuanto el flujo generado por el usuario es comparativamente menor al recibido desde el proveedor de servicios; por tanto, no es necesario que el canal de retorno sea dotado de una gran capacidad. Sin embargo, estos canales no siempre son administrados por una misma compañía operadora, lo que limita la disponibilidad y calidad del servicio prestado.

Frente a esto y como consecuencia de la gran demanda por parte de los usuarios a la integración de servicios interactivos caracterizados por tráfico simétrico y asimétrico en un solo sistema de difusión, se busca que los sistemas

satelitales satisfagan estas necesidades prescindiendo del canal de retorno terrestre.

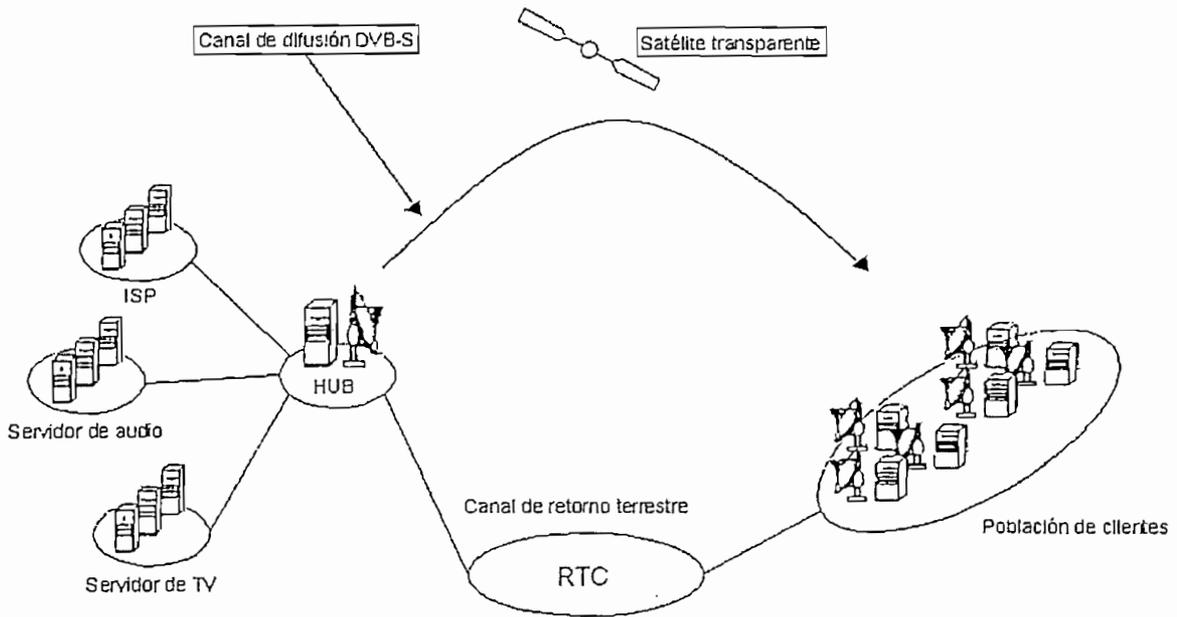


Figura 1.17 Esquema general de un sistema satelital de servicios múltiples.

Dentro de los sistemas de comunicación por satélite, solo aquellos que utilizan satélites GEO son capaces en la actualidad de ofrecer servicios de banda ancha y es precisamente el estándar DVB el de mayor utilización y que forma parte de numerosos proyectos actuales y futuros que incluyen la especificación para el canal de retorno también satélite DVB-RCS, formando de esta manera redes satelitales multiservicio completamente interactivas. [7]

Por estos motivos es importante realizar una descripción de los principales sistemas satelitales de banda ancha basados en la tecnología DVB, mencionando también a aquellos que incorporan satélites con funciones OBP-MR (*On-Board Processing MPEG-2 Remultiplexing*) cuyas ventajas están relacionadas por un lado, con la distribución de los proveedores de contenidos, y por otro, con ampliar el campo de aplicabilidad hacia servicios multimedia interactivos de tiempo real como los de VoIP o videoconferencia.

Dentro de los sistemas satelitales que se basan en la tecnología DVB y que además incorporan satélites con OBP-MR, se encuentran: Skyplex de Eutelsat [8], IBIS de Alcatel [9] y SpaceMux de EMS [10] como los de mayor relevancia para el presente proyecto.

1.2.1 EL SISTEMA SATELITAL SKYPLEX [8]

La propuesta satelital multimedia de Eutelsat es el sistema Skyplex que utiliza los satélites de la clase *Hot Bird*. El principal concepto de Skyplex es el de proveer un sistema distribuido de enlaces ascendentes; es decir, el sistema permite la difusión de flujos de transporte MPEG-2 (*Motion Picture Experts Group*) con contenidos de programas de televisión, multimedia o Internet desde localidades independientes, generalmente dispersas geográficamente (ver figura 1.18).

El sistema multiplexa a bordo del satélite los diferentes flujos de transporte y los transmite a través de una única señal en formato DVB-S, de tal forma que puedan ser recibidos por los tradicionales receptores IRD (*Integrate Receive Decoder*) o computadores equipados con tarjetas receptoras. La señal DVB-S que reciben los decodificadores no guarda ninguna diferencia con un canal multiplexado en tierra y transmitido por un satélite GEO transparente.

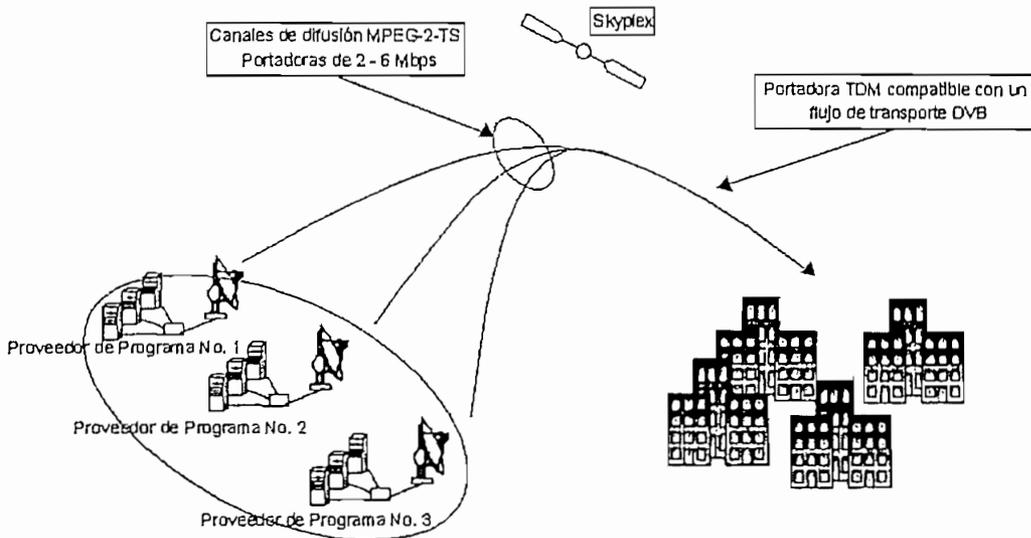


Figura 1.18 Arquitectura del sistema Skyplex.

Como se puede observar en la figura 1.18, los esquemas de transmisión para los canales ascendentes y descendentes son diferentes y se detallan a continuación:

- En el enlace ascendente, las estaciones pueden acceder al sistema de forma continua o exclusiva en un modo de acceso denominado SCPC (*Single Channel per Carrier*), o utilizando acceso TDMA (*Time Division Multiple Access*). Ambos tipos de acceso tienen una tasa máxima de transmisión en el orden de los 6 Mbps.
- En el enlace descendente, existe una única señal de salida transmitida por el satélite que corresponde a un flujo de transporte TDM a 27.5 Msymb/s (38 Mbps con un FEC de tasa 3/4).

Dentro del sistema Skyplex, la generación de los paquetes de transporte, se realiza de forma independiente en tierra para cada enlace ascendente. Los paquetes MPEG-2 que contienen vídeo, audio o datos se multiplexan en tierra en el múltiplex de transporte (ver figura 1.19). De esta forma un simple multiplexor se encuentra a bordo del satélite el cual crea un único enlace descendente en formato DVB-S de 27,5 Msymb/s.

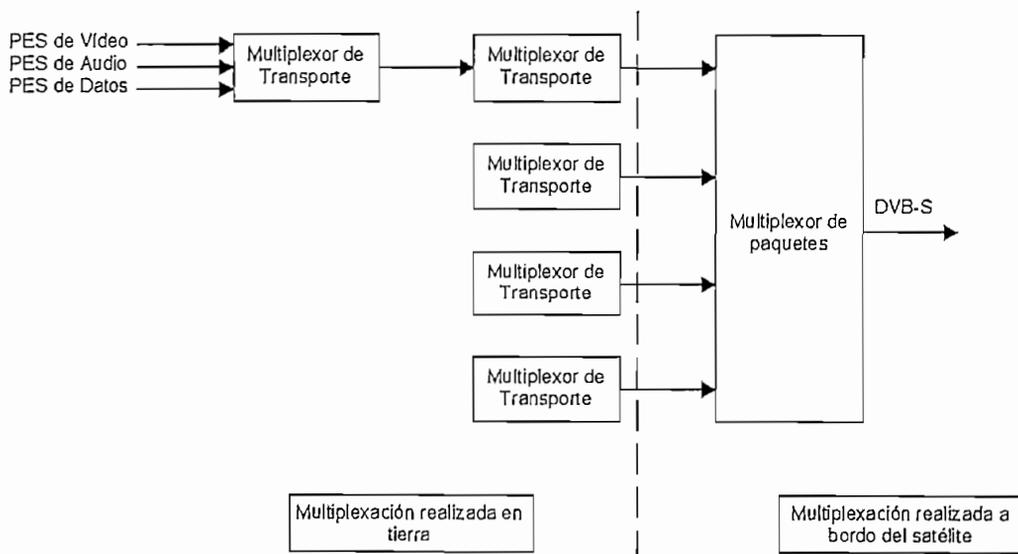


Figura 1.19 Arquitectura de multiplexación del sistema Skyplex.

Puede notarse que el sistema Skyplex resuelve el problema de la distribución de los proveedores de servicios mediante la utilización de funciones de remultiplexación MPEG-2 a bordo del satélite. Sin embargo, los sistemas actualmente operativos Skyplex son básicamente unidireccionales y la interacción para los servicios interactivos, como el acceso a Internet, se consigue a través de canales de retorno terrestre con los inconvenientes anotados anteriormente.

La orientación del sistema, al menos en las unidades Skyplex de los *Hot Bird* 4 y 5, es la de prestar servicios principalmente de difusión y por esta razón no se abordan temas como los mecanismos para proveer garantías de calidad en los servicios interactivos o el tipo de aplicaciones a ser soportadas.

Otro inconveniente del sistema Skyplex es la utilización de canales ascendentes no estandarizados, aunque la nueva generación de satélites Skyplex, representada por el *Hot Bird* 6, presentará compatibilidad con la interfaz aérea del estándar de canal de retorno DVB-RCS.

Se pretende que en las futuras versiones del sistema Skyplex esta compatibilidad se extienda al modelo de red y señalización, de tal forma que el sistema utilice estaciones bidireccionales DVB-RCS estándar.

1.2.2 EL SISTEMA SATELITAL IBIS[9]

La propuesta del sistema IBIS (*Integrated Broadcast Interaction System*) cuenta con un satélite con procesamiento a bordo (Alcatel 9343) completamente regenerativo diseñado para proveer acceso distribuido y directo, compatible con el estándar DVB-RCS.

El sistema está orientado a dar servicio a programadores de difusión individual de vídeo digital, proveedores de acceso a Internet (*ISP*) y usuarios multimedia. El esquema general del sistema IBIS se muestra en la figura 1.20.

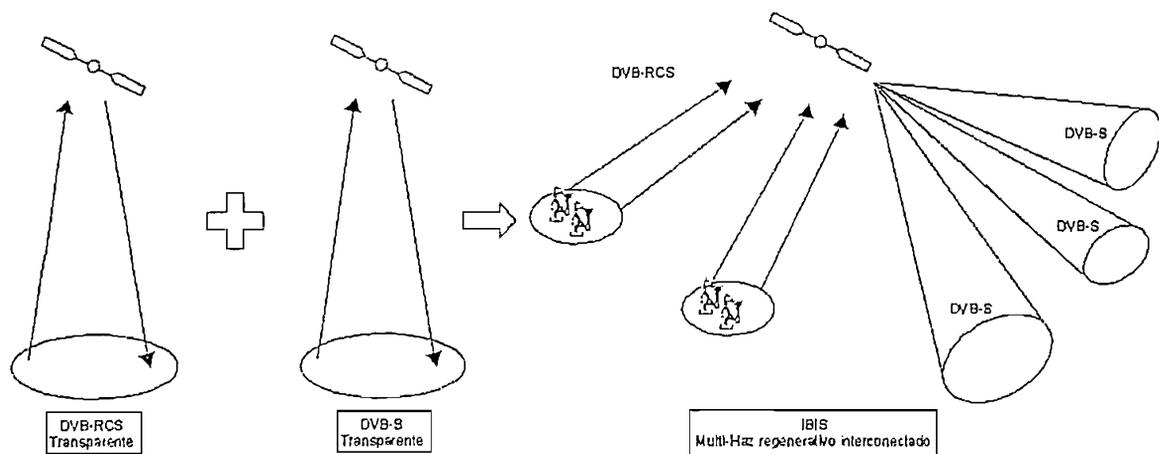


Figura 1.20 Concepto del sistema IBIS de Alcatel.

El sistema IBIS integrará los formatos para el canal de difusión DVB-S y retorno DVB-RCS de la tecnología DVB. En la actualidad, la mayoría de sistemas que utilizan esta tecnología emplean satélites GEO transparentes sin ningún tipo de procesamiento a bordo.

IBIS combinará los dos formatos dentro de un mismo sistema satelital regenerativo y multihaz completamente interconectado (*OBS, On-Board Switching*). Esta característica hace que los usuarios empleen estaciones RCSTs (*Return Channel Satellite Terminal*) estándares, de las que se espera su utilización y producción a gran escala de tal forma que su precio sea relativamente reducido.

La red IBIS está orientada a dar servicio a una gran variedad de usuarios, que van desde el mercado doméstico hasta los usuarios de grandes corporaciones. Los usuarios individuales y los proveedores de servicios podrán acceder al sistema mediante cualquiera de los haces iluminados por el satélite.

El acceso al sistema se realiza usando un conjunto de múltiples frecuencias dentro de una trama TDMA (*MF-TDMA, Multiple Frequency - Time Division Multiple Access*) donde existen diferentes tasas de transmisión de

acuerdo al tipo de estación RCST tal como se especifica en el estándar DVB-RCS y que será estudiado más adelante.

Tal como se muestra en la figura 1.20, el satélite realiza la remultiplexación y cambio de formato de los enlaces ascendentes DVB-RCS en un solo enlace descendente que es totalmente compatible con el estándar DVB-S. La remultiplexación que realiza el satélite del sistema permite que las estaciones de usuario cuenten con receptores DVB-S los cuales son accesibles al mercado del usuario común, tienen precios relativamente reducidos, cuentan con buen rendimiento y están ampliamente extendidos en Europa y América del Norte.

Una característica importante del sistema es la capacidad de encaminar los datos que pertenecen a un enlace ascendente de cualquier haz hacia cualquier combinación de haces en el enlace descendente. El sistema implementa una interconectividad total entre cualquier enlace ascendente DVB-RCS y el enlace descendente DVB-S.

A fin de conseguir esta interconectividad todos los enlaces ascendentes DVB-RCS tienen que ser demultiplexados, demodulados y decodificados para después ser conmutados y remultiplexados en el canal descendente DVB-S, según sean las demandas de servicios de los usuarios.

El modelo de referencia DVB-RCS para redes interactivas por satélite requiere dos saltos satelitales para la comunicación entre estaciones remotas RCSTs. Esto significa que el retardo de propagación es lo suficientemente significativo para excluir aplicaciones interactivas multimedia, como los servicios de VoIP o de videoconferencia. El sistema IBIS permitirá comunicaciones en un solo salto entre terminales remotos en servicios punto a punto, punto a multipunto o multipunto a multipunto.

Como puede observarse, el sistema IBIS aborda los problemas de dispersión de los proveedores de servicio (multiservicio multisitio) y utiliza un canal de retorno DVB-RCS para las estaciones remotas, de esta forma el sistema

satelital es completamente interactivo y se basa en la tecnología DVB para el canal de difusión y retorno.

Sin embargo, el sistema IBIS es un proyecto a futuro y uno de los primeros en utilizar satélites con remultiplexación MPEG-2, conmutación a bordo y además es compatible con la tecnología DVB, por lo que existen varios aspectos de diseño que deben ser resueltos y que no se encuentran especificados o están abiertos dentro de la norma DVB-RCS.

1.2.3 EL SISTEMA SATÉLITE SPACEMUX [10]

En la misma línea que el sistema IBIS, en cuanto a la utilización de satélites con OBP-MR y funciones OBS, se encuentra el sistema SpaceMux de EMS Technologies. Al igual que en el caso anterior el SpaceMux ofrecerá la remultiplexación, conmutación entre distintos haces y cambio de formato de los enlaces ascendentes DVB-RCS en un solo enlace descendente DVB-S.

El principal objetivo del sistema radica en la posibilidad de ofrecer aplicaciones interactivas en un solo salto, es decir, conexiones en malla entre estaciones remotas de pequeño tamaño y bajo precio. Estas aplicaciones incluyen servicios de conferencia y comercio entre negocios (B2B *business to business*).

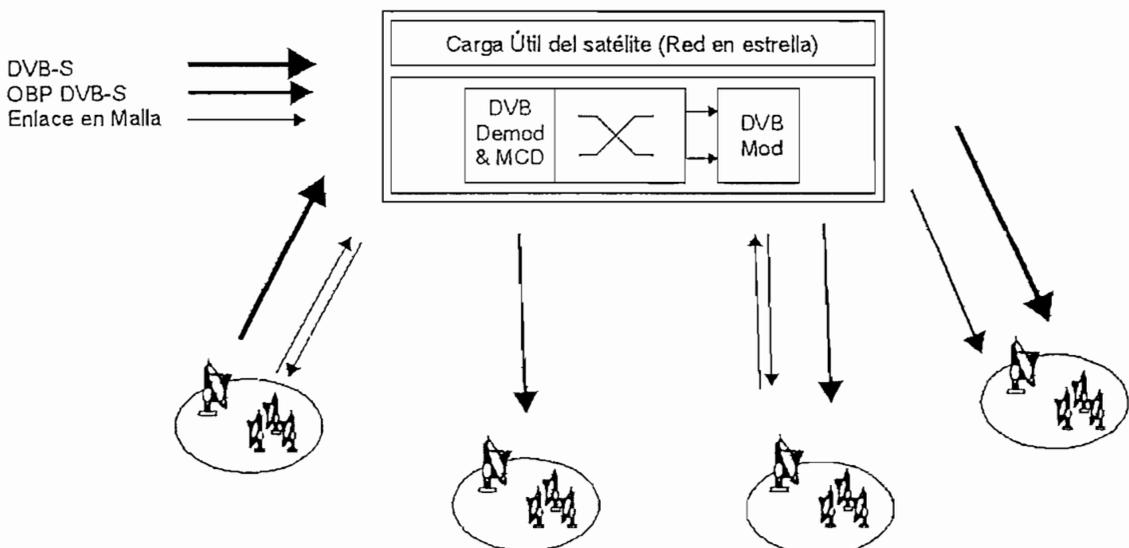


Figura 1.21 Arquitectura general del sistema SpaceMux.

La arquitectura general del sistema SpaceMux puede observarse en la figura 1.21. La representación muestra un conjunto de redes satelitales en estrella cubiertas por diferentes haces; además, el sistema tiene la capacidad de manejar redes en estrella DVB-S transparentes así como también sistemas que utilizan la capacidad de remultiplexación.

Puede notarse que el canal DVB-S remultiplexado puede transmitirse de forma simultánea en diferentes haces por lo que el procesador a bordo tiene que realizar funciones de conmutación, además de las de conversión de formato y regeneración de la señal.

Las portadoras DVB-RCS ascendentes son de 2,048 Mbps y están agrupadas en función del tipo de estrella en la que se utilizan (transparente o con remultiplexación a bordo).

El control de la asignación de recursos se realiza mediante un nodo central que puede ser una estación dedicada o localizada en la misma ubicación del *hub* de la red. Las funciones de este nodo incluyen el procesamiento de la señalización del canal de difusión y retorno, la programación de la asignación de recursos, funciones de admisión de llamadas y de facturación.

El nodo de control cuenta con una conexión dedicada con la unidad de proceso a bordo, de tal forma que la interacción entre estos dos elementos es transparente a los terminales de usuario. El nodo de control podría eventualmente realizar las funciones de pasarela para conexiones de conferencia con el canal de retorno terrestre o RTC.

El enlace descendente sigue el formato DVB-S de tal forma que es posible equipar a los terminales con receptores de bajo precio.

Es importante además mencionar que EMS *Technologies* se encuentra desarrollando investigaciones a fin de implantar, no solamente las funciones de remultiplexación MPEG-2 y conmutación a bordo, sino también la incorporación

del manejo de la planificación de recursos por parte de la unidad de proceso, de esta forma se pretende evitar el doble salto en petición y respuesta en la asignación de recursos. Es decir, la unidad de proceso a bordo incorpora las funciones de planificación que realiza el nodo central de control mejorando en un cuarto de segundo la respuesta a una petición hecha por las RCSTs. [11]

De la misma forma que en el caso del sistema IBIS, el sistema SpaceMux es un proyecto a futuro y su versión de demostración todavía se encuentra en pruebas operativas.

Uno de los aspectos a destacar es el hecho de que este sistema junto con IBIS y Skyplex permiten identificar una tendencia tecnológica hacia la utilización de funciones de remultiplexación MPEG-2, conmutación entre distintos haces y cambio de formato de los enlaces ascendentes para habilitar los servicios interactivos en un solo salto y permitir la distribución de los proveedores de servicio; todo esto basado en la plataforma DVB.

1.3 REDES SATELITALES VSAT BASADAS EN LA TECNOLOGÍA DVB-S/RCS

Un aspecto que ha marcado un punto de inflexión en la industria VSAT en los últimos años, ha sido la amplia aceptación del estándar de difusión de vídeo digital para satélite DVB-S.

La especificación permite la construcción de sistemas en principio unidireccionales y que se han utilizado principalmente en servicios de difusión de canales de televisión digital. Estos sistemas satelitales son los primeros orientados al mercado del usuario común o doméstico, de allí que en consecuencia los equipos VSAT experimenten una importante reducción de precio y tamaño.

Sin embargo, la demanda de servicios de los usuarios también ha experimentado cambios, debido por ejemplo al desarrollo del Internet, que ha motivado la transformación de estos sistemas satelitales unidireccionales en sistemas de comunicación bidireccionales, aunque en principio esta interacción se alcanzaba mediante el uso de canales de retorno terrestre, tal como se mencionó anteriormente.

Existen también sistemas que utilizan canales de retorno satelital propietarios que pertenecen a las dos compañías más importantes de la industria VSAT que son: *DIRECWAY Two-way System* de *Hughes Network Systems* [12] y *StarBand* de *Gilat* [13].

Los sistemas de difusión DVB-S han podido afrontar los cambios en las necesidades de los usuarios por lo que el desarrollo tecnológico ha sido continuado hasta alcanzar la aprobación de la especificación del canal de retorno DVB-RCS y la inclusión de satélites con funciones de remultiplexación, conmutación y cambio de formato de los enlaces ascendentes a bordo. Gracias a estos aspectos, los sistemas de comunicación vía satélite orientados al mercado del usuario común y basados en la tecnología DVB, pueden ser en la actualidad totalmente interactivos por satélite.

Actualmente las redes satelitales VSAT no se muestran únicamente como una solución en zonas de limitada o pobre infraestructura terrestre y tampoco están orientadas solamente a prestar servicio a pequeñas o grandes empresas mediante sistemas de comunicación propietarios. Las redes VSAT tienen en estos momentos la capacidad de competir en precio y servicios con los sistemas de comunicación terrestre y son capaces de llegar directamente a la ubicación del usuario común. [6]

En este punto cabe recalcar que la tendencia tecnológica de las redes VSAT se marca por la utilización del estándar DVB-RCS para el canal de retorno satelital, el cual establece las pautas para la construcción de sistemas multimedia completamente interactivos por satélite ICMSN (*Interactive Channel for*

Multimedia Satellite Networks) [6] con la ventaja de que esta especificación se basa en la tecnología DVB cuya aceptación respecto al canal de difusión DVB-S ha sido amplia.

1.3.1 REDES SATELITALES VSAT DVB MULTIMEDIA INTERACTIVAS [6]

Una red satelital multimedia interactiva ICMSN es un sistema de comunicación totalmente vía satélite que provee interconexión entre usuarios que intercambian aplicaciones de tiempo real basadas en diferentes tipos de datos (texto, voz, imágenes, vídeo, etc.).

Un sistema ICMSN consiste de una estación *hub* en tierra uno o más satélites para la difusión, un conjunto de estaciones VSAT interactivas RCSTs y un satélite para el canal de retorno.

La figura 1.22 describe el principio de funcionamiento de una red ICMSN. Como se puede observar, el tipo de usuarios incluye pequeños y medianos comerciantes, universidades y usuarios residenciales.

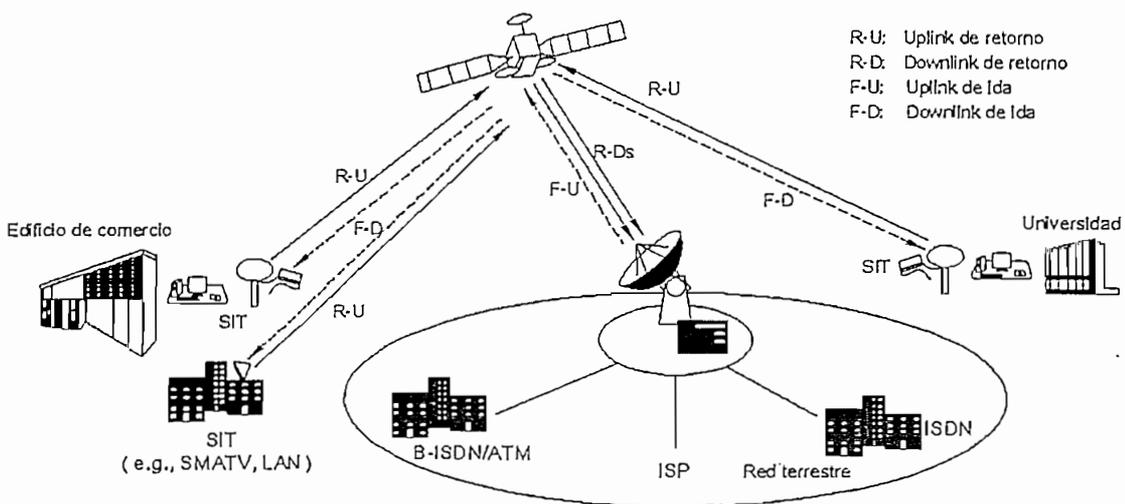


Figura 1.22 Principio de funcionamiento de una red ICMSN.

El tráfico transmitido desde el *hub* central hacia las RCSTs de los usuarios es multiplexado dentro de un flujo de difusión convencional DVB-S.

El canal de retorno por su parte, será operado como parte de la misma red digital, en donde el *hub* provee una puerta de salida o *gateway* hacia otras redes.

En función de mantener una interconectividad con otras redes, se utilizarán estándares ampliamente difundidos para transmitir datos desde las RCSTs hacia el *hub*, en particular IP y ATM.

El canal de retorno es provisto desde una RCST, la cual posee una pequeña antena (0,5 m a 1,2 m) y un computador personal multimedia o un IRD hacia un servidor interactivo en una estación *hub* usando un esquema de múltiple acceso.

Está previsto que una o más RCSTs estén ubicadas en las propiedades de los clientes con múltiples computadores personales conectados a través de una LAN.

Se espera que hasta $\frac{1}{4}$ de millón de RCSTs sean soportadas por una simple ICMSN. Con múltiples computadores detrás de cada RCSTs se podría tener un número máximo de *hosts* de aproximadamente 8 millones en una sola red.

1.3.2 VENTAJAS Y DESVENTAJAS FRENTE A ESQUEMAS DE UTILIZACIÓN ACTUALES

1.3.2.1 Ventajas

Las redes satelitales VSAT DVB Multimedia Interactivas cuentan con características inherentes que no se encuentran o son de muy difícil aceptación en otro tipo de redes, como por ejemplo las terrestres. Entre las características más sobresalientes se pueden anotar las siguientes:

- Ubicuidad, dentro del área de cobertura del satélite. Mediante esta característica y el uso de pequeños terminales bidireccionales VSAT se puede llegar directamente al usuario común y solucionar el problema de acceso a la última milla.
- Independencia de la infraestructura de comunicación terrestre habilitada. Los sistemas satelitales VSAT DVB Multimedia Interactivos solamente necesitan un punto de visión al satélite GEO que presta el servicio. Esta característica permite dar un salto cualitativo importante en zonas donde no existe una infraestructura de comunicación terrestre o ésta es muy limitada.
- Independencia de la densidad poblacional de la zona de servicio. Toda infraestructura de comunicación terrestre necesita contar con un número determinado de clientes potenciales para hacer que la inversión sea rentable. En las comunicaciones satelitales basta con que el cliente se encuentre dentro de la zona de cobertura para que pueda acceder al servicio.
- Asimetría de flujos de información. Las redes satelitales VSAT DVB Multimedia Interactivas muestran por naturaleza una gran asimetría entre la capacidad del canal de difusión (*outbound*) y los canales de retorno de los clientes remotos (*inbounds*). Esta asimetría es una característica común en servicios de alta demanda, como por ejemplo, acceso a Internet, donde el usuario recibe mucha más información de la que envía.
- Comunicación punto a multipunto y punto a punto. La capacidad de difusión (*broadcast*) es inherente en los sistemas satelitales, esta difusión puede ser adaptable fácilmente a *multicast*, característica que es de difícil utilización en redes terrestres.
- Rápida instalación de la red.
- Los costos de conexión son independientes de la distancia, dentro de la zona de cobertura del satélite. Los costos del servicio son a menudo inferiores que

las alternativas terrestres para las mismas velocidades de transmisión y con amplia distribución geográfica de las localidades. [14]

- La flexibilidad de la red, capacidad de crecimiento y reconfiguración. Los terminales VSAT son de fácil instalación, remoción y desplazamiento.

1.3.2.2 Desventajas

Dentro de las principales desventajas que encierra una comunicación con redes VSAT DVB Multimedia Interactivas se pueden anotar las siguientes:

- El enlace de radiofrecuencia está sujeto a interferencia como resultado del reducido tamaño de la antena que utilizan.
- La interceptación de información no autorizada debido a la gran zona de cobertura del satélite. La encriptación de la información resulta casi obligatoria.
- Retardo satélite. El retardo de propagación de un salto satélite oscila entre 240 y 280 milisegundos dependiendo de la latitud y la diferencia de longitudes entre los terminales y el satélite. En una configuración típica en estrella VSAT una comunicación entre clientes remotos implica un doble salto, es decir un retardo de propagación de aproximadamente medio segundo, circunstancia que limita el tipo de aplicaciones que se pueden ofrecer en un sistema, especialmente multimedia con fuertes restricciones de retardo (videoconferencias o llamadas de voz sobre IP).
- Aunque la instalación de los sistemas VSAT DVB se ha reducido en complejidad, todavía es necesaria de la intervención de un personal técnico especializado, a diferencia de algunos sistemas de comunicación terrestre de banda ancha donde la tendencia está en que sea el usuario final quien realice este proceso a efectos de reducir los precios de contratación.

Referencias Bibliográficas del Capítulo 1

- [1] G. Maral, "VSAT Networks", John Wiley & Sons, England 2003.
- [2] http://www.upv.es/satelite/trabajos/pract_4/vsat_hpg.htm
- [3] http://www.es.northropgrumman.com/SATCOM/vsat/images/vsat_new.jpg
- [4] http://www.telemann.co.jp/img/ka_image03.jpg
- [5] <http://www.superbird.co.jp/services/img/idu-omote.jpg>
- [6] J. Neale, R. Green, J. Landovski, "Interactive Channel for Multimedia Satellite Networks", IEEE Communications Magazine, Marzo 2001.
- [7] D. Bem, T. Wieckowski, R. Zielinsky. "Broadband Satellite Systems", IEEE Communications Surveys & Tutorials, Año 2000. Volumen 3 Número 1.
- [8] H. T. Fenech, A. Pujante-Cuadrapani and E. Lance "Eutelsat Multimedia Satellites", Space Communications – An International Journal. Volumen 17 Números 1-3, Junio 2001.
- [9] S. Chacón, J. Casas, A. Cal, et al, "Multimedia Applications of the Integrated Broadcast Interaction System (IBIS)", ESA Workshop on Digital Signal Processing, Lisbon (Portugal), Octubre 2001.
- [10] P. Takats, and M. Coté, "SpaceMuxTm: An On-Board Mesh Processor", Space Communications – An International Journal. Volumen 17 Números 1-3, Junio 2001.
- [11] M. Coté, C. Black, and A. Iuoras, "On-Board Scheduling for Multimedia Applications", Space Communications – An International Journal, Volumen 17 Números 1-3, Junio 2001.
- [12] <http://www.directpc.com>.
- [13] <http://www.starband.com>
- [14] D. Azaren, W. Courtney, J. Freitag et al., "The Multimedia Migration: Transponder Versus Processing Payload VSAT Networks", Sixth Ka-Band Utilization Conference, Cleveland Ohio, USA. Mayo – Junio 2000.

CAPÍTULO 2

ESTÁNDARES DVB-S/RCS

En el capítulo anterior se expuso el estado del arte de las redes satelitales VSAT y se revisó brevemente algunos de los sistemas que contemplan la implementación de los estándares DVB-S/RCS como una alternativa viable y rentable para la creación de redes totalmente interactivas vía satélite.

Se pudo notar la existencia de una tendencia marcada por parte de las redes VSAT hacia la utilización de los estándares DVB-S/RCS para soportar los requerimientos que exigen las aplicaciones multimedia actualmente.

Es precisamente por esta tendencia que el presente capítulo está dedicado al estudio de dichos estándares, enfatizando en aquellos conceptos y recomendaciones propios de las normas que permitirán más tarde modelar "La Red para Acceso a Internet", objetivo del presente proyecto.

Como se explicará más adelante en este mismo capítulo, tanto DVB-S como DVB-RCS utilizan como medio de multiplexaje y transporte de datos al sistema de codificación MPEG-2 (*Motion Picture Experts Group*).

Por esta razón se partirá de una revisión de aquellos conceptos del sistema MPEG-2 que permitan entender cómo la información DVB-S/RCS puede ser multiplexada y transportada por el mismo.

Se continuará el capítulo con el estudio mismo de los estándares del canal de difusión DVB-S y del canal de retorno DVB-RCS vía satélite, revisando todas las recomendaciones y reportes técnicos que permitan tener una idea global de los mecanismos, elementos y procedimientos contemplados por las normas para armar una red satelital interactiva bajo la plataforma DVB-S/RCS.

2.1 EL SISTEMA MPEG-2 [1]

MPEG-2 es el nombre que se le ha dado a un grupo de estándares de codificación de audio y vídeo acordado por MPEG (grupo de expertos en imágenes en movimiento), y que han sido publicados como una norma de la ISO bajo la denominación ISO/IEC 13818-1.

Este sistema ha sido típicamente utilizado para la transmisión de señales codificadas que incluye transmisiones Satelitales Digitales y de *Cable TV*. Con algunas modificaciones MPEG-2 es también el formato de codificación que se usa, por ejemplo, en los discos SVCDs y DVDs, y es a su vez contemplado como medio de transporte para los sistemas DVB-S/RCS por ser uno de los formatos de compresión más sofisticados y de mayor aceptación dentro de la industria de las comunicaciones.

Es así que en el conjunto de estándares DVB-S y DVB-RCS se define a la codificación de fuente de la señal como una adaptación del sistema MPEG-2. En otras palabras, la señal de entrada y de salida específica para el sistema DVB-S/RCS es la denominada "MPEG-2 *Transport Stream*" (MPEG-2 TS), descrita más adelante en este mismo capítulo.

En esta sección entonces, se describirán ciertas características del sistema MPEG-2, principalmente aquellas referentes al flujo de transporte TS, tomando en consideración lo expuesto en ISO/IEC 13818-1 y ahondando en los temas de interés para el posterior desarrollo del presente proyecto.

2.1.1 FORMACIÓN DE LOS FLUJOS DE SEÑAL MPEG-2

En la figura 2.1 se muestra, a manera de un diagrama de bloques, el proceso por el que atraviesa un programa o conjunto de programas para dar lugar a los flujos de señal MPEG-2. Se debe entender, según la terminología de MPEG, como "Programa" a un servicio o canal simple radiodifundido.

Dicho proceso puede ser dividido en dos grandes bloques o capas, con el fin de facilitar su entendimiento: La Capa de Compresión y la Capa de Sistema. En la Capa de Compresión se realizan las operaciones propiamente dichas de codificación MPEG, recurriendo a los procedimientos generales de compresión de datos, audio y vídeo.

En la Capa de Sistema, por su parte, se llevan a cabo los procesos que conducen a la obtención de los flujos de señal MPEG-2 consistentes en la organización en paquetes de los datos comprimidos y el posterior multiplexado de todas las señales asociadas a un programa o programas (datos, audio, vídeo, etc) en un solo flujo de datos.

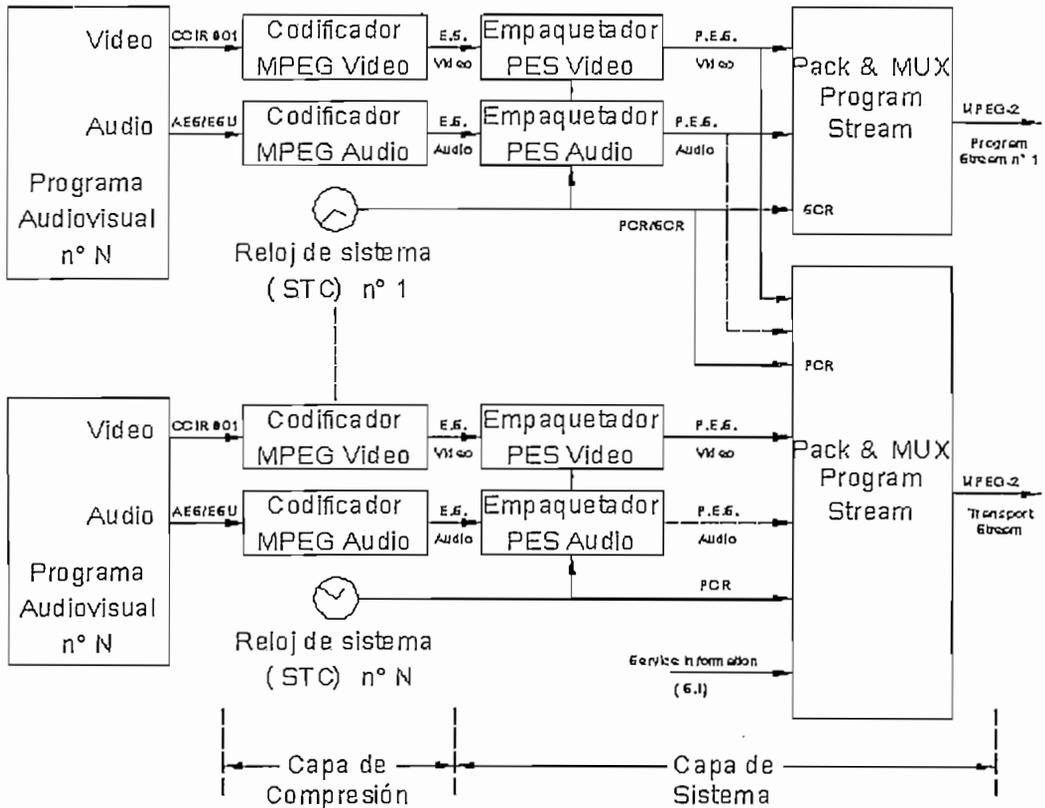


Figura 2.1 Esquema conceptual de la formación de flujos de señal MPEG-2.

En la figura 2.1 se puede observar que el componente básico de un flujo MPEG son los flujos elementales o ES (*Elementary Stream*), los cuales corresponden a señales de vídeo, audio y datos de un programa, que

generalmente se encuentran comprimidas luego de haber sido procesadas por la Capa de Compresión.

Estos ESs pasan a continuación a la Capa de Sistema donde son convertidos en un flujo elemental paquetizado o PES (*Packetized Elementary Stream*). Un flujo PES está compuesto íntegramente por paquetes PES los cuales son generalmente de longitud variable, y normalmente se los limita a 64 *Kbytes*.

En el sistema MPEG-2, los PES pueden ser multiplexados de dos maneras diferentes formando así dos tipos distintos de flujos, llamados flujo de programa PS (*Program Stream*) y el otro flujo de transporte TS (*Transport Stream*), siendo este último el de mayor relevancia para el presente estudio por ser el que se utiliza como medio de transporte en el sistema DVB-S/RCS. [2]

2.1.2 FLUJO DE PROGRAMA PS (*PROGRAM STREAM*)

El flujo de programa PS es el resultado de combinar uno o más PES, los cuales pertenecen a un mismo Programa (servicio o canal) y que comparten obligatoriamente el mismo reloj de referencia (ver figura 2.1). El PS es un flujo que ha sido diseñado para transmitir o almacenar un programa de datos codificados en un ambiente relativamente libre de errores, y donde el proceso que realiza el sistema de codificación (que puede ser por ejemplo a través de software), es de importante consideración [2].

Por su vulnerabilidad en ambientes susceptibles a errores y por su limitación de transportar información de un solo programa, el PS no está considerado para ser el medio por el cual los datos DVB-S/RCS sean transportados, no así el TS que será visto con mayor detenimiento a continuación.

2.1.3 FLUJO DE TRANSPORTE TS (*TRANSPORT STREAM*)

El flujo de transporte TS conocido también como “Múltiplex de Transporte TS”, ha sido diseñado para la transmisión y almacenaje de datos codificados en

ambientes que pueden introducir errores significativos. Tales errores pueden ser manifestados como bits errados o paquetes perdidos. [2]

El TS puede ser construido a partir de cualquier método que resulte en un flujo válido. Es posible construir TSs que contengan uno o más programas de flujos elementales de datos codificados, de PSs o de otros TSs los cuales pueden contener a su vez uno o más programas.

EL múltiplex TS está compuesto íntegramente por “paquetes de transporte” o “paquetes MPEG-2 TS”, los cuales deben tener una longitud relativamente corta para introducir métodos eficaces de corrección de errores.

Cada paquete MPEG-2 TS tiene una longitud de 188 *bytes* precedido de un campo de adaptación. Esta longitud fue establecida especialmente para emisiones vía satélite, enlaces de cable o terrestre contemplados en la norma europea DVB.

Los paquetes de transporte se forman a partir de los “paquetes PES” correspondientes a cada “Flujo Elemental (ES)” de señal (vídeo, datos, audio, etc.), tal como se muestra en la figura 2.2.

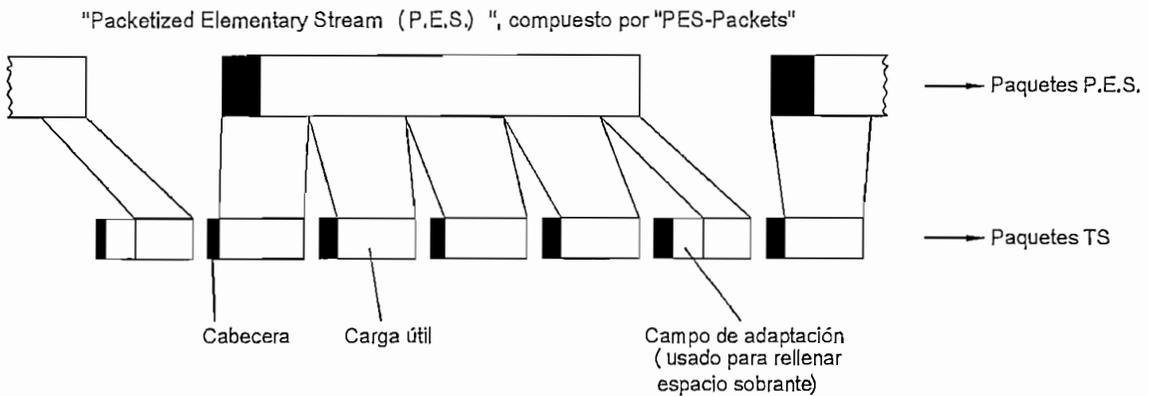


Figura 2.2 División de un PES en una serie de paquetes MPEG-2 TS.

El proceso de formación de los paquetes de transporte está sujeto al cumplimiento de dos condiciones fundamentales que se muestran a continuación:

- Un paquete de transporte solo puede contener datos tomados de los paquetes PES.
- El primer *byte* de cada paquete PES debe ser el primer *byte* del *payload* de un paquete de transporte.

2.1.3.1 Formación del Múltiplex de Transporte TS

Luego de ser aplicado el proceso anterior a cada uno de los ESs pertenecientes a su vez a varios programas, los paquetes MPEG-2 TS resultantes se disponen secuencialmente para conformar el “Múltiplex MPEG-2 TS”.

Cabe indicar que la norma no presenta restricciones en cuanto al orden en que los paquetes deben ser multiplexados dentro del TS; simplemente debe respetarse el orden cronológico de los paquetes de transporte pertenecientes a un mismo ES.

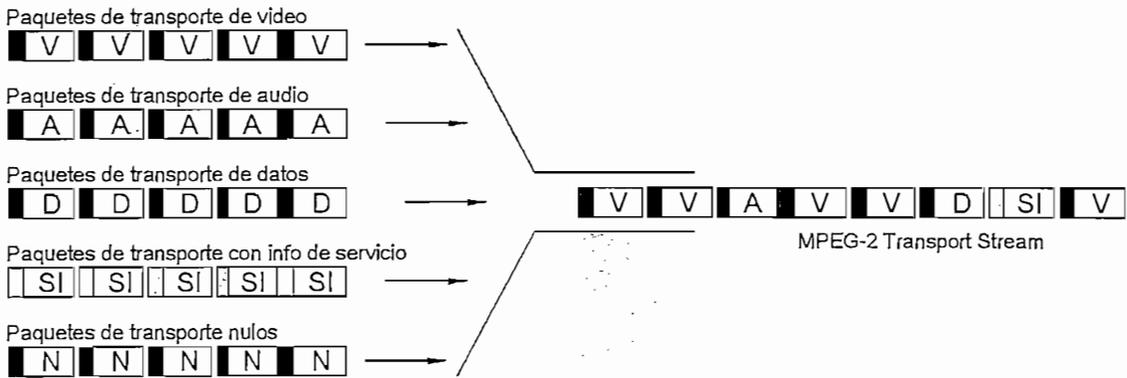


Figura 2.3 Conformación del Múltiplex de Transporte TS.

Es importante señalar también que, además de los paquetes de transporte asociados a los ESs, en MPEG es necesario incorporar paquetes de transporte que contienen información sobre los distintos programas (esto es a través de la PSI que será revisada más adelante), así como paquetes de transporte nulos que

se emplean para absorber eventuales reservas de capacidad del múltiplex. La figura 2.3 muestra de una forma simplificada, cómo el proceso de conformación del Múltiplex de Transporte TS se lleva a cabo.

2.1.3.2 Estructura del paquete de transporte MPEG-2 TS

Como ya se mencionó, el flujo de transporte TS está conformado por paquetes MPEG-2 TS de tamaño fijo. Dichos paquetes de 188 bytes se componen de una cabecera de 4 bytes y de una carga útil de 184 bytes como máximo, tal como se indica en la figura 2.4.

Campo	Definición	No. de Bits
sync_byte	Byte de sincronización 01000111 (0x47H)	8
transport_error_indicator	Identifica un error detectado en la transmisión anterior	1
payload_unit_start_indicator	Inicio de PES en el paquete de transporte	1
transport_priority	Indicador de prioridad	1
PID	Identificador del paquete de transporte	13
transport_scrambling_control	Tipo de cifrado de transporte	2
adaptation_field_control	Control del campo de adaptación en el paquete	2
continuity_counter	Contador de continuidad entre paquetes afines	4

Tabla 2.1 Estructura de la cabecera de un Paquete MPEG-2 TS.

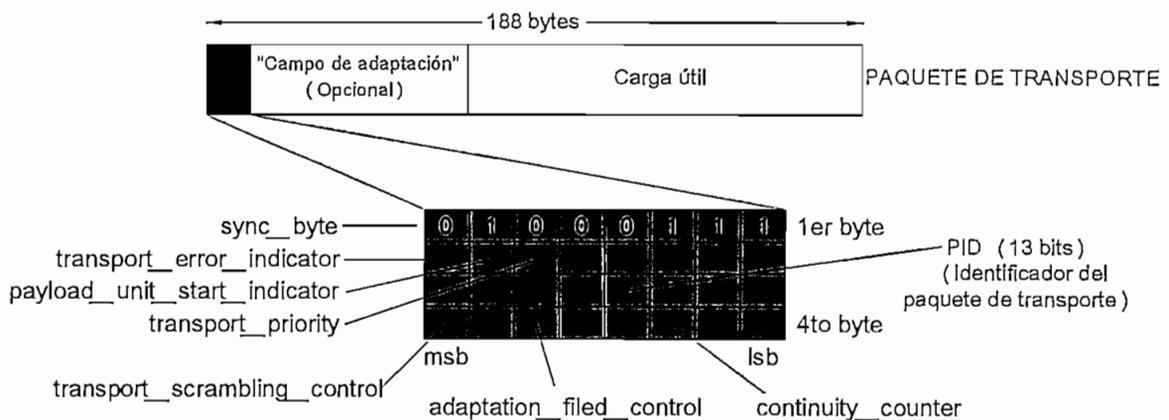


Figura 2.4 Paquete de transporte MPEG-2.

La cabecera del paquete de transporte MPEG-2 tiene un tamaño fijo e inalterable de 4 *bytes*. El formato de la misma se muestra en la tabla 2.1.

A continuación se detalla cada uno de los campos presentes en la cabecera de los paquetes MPEG-2 TS mostrados en la tabla 2.1.

- *sync byte* (Byte de sincronización)

Es el primer *byte* de la cabecera MPEG-2 y constituye un campo fijo de 8 bits cuyo valor es 0x47 (01000111).

- *transport error indicator* (Indicador de error de transporte)

El indicador de error de transporte es básicamente un *bit* de bandera. Cuando su valor es '1' indica que se detectó un error en la transmisión anterior; caso contrario permanecerá en '0'.

- *payload unit start indicator* (Indicador de inicio de carga útil)

El indicador de inicio de carga útil es también un bit de bandera. Se fija en '1' cuando la carga útil contiene el inicio de un PES. Los siguientes paquetes de un mismo PES tendrán el valor de '0', al igual que para los paquetes nulos. El significado de esta bandera cuando el paquete transporta datos privados no está definido en la recomendación MPEG.

- *transport priority* (Prioridad de transporte)

Es un *bit* indicador que cuando se fija en '1' indica que el paquete asociado es de mayor prioridad que otros paquetes con el mismo PID que no tienen este *bit* en '1'. El mecanismo de transporte puede usar este *bit* para priorizar sus datos dentro de un flujo elemental. Dependiendo de la aplicación este campo puede ser fijado sin importar el PID.

- Paquet Identifier (Indicador de paquete PID)

El PID es un campo de 13 bits que indica el tipo de datos presentes en la carga útil del paquete MPEG-2 TS. Como se explicó anteriormente en un TS pueden estar multiplexados varios programas diferentes, cada uno de los cuales contiene a su vez muchos flujos elementales distribuidos en paquetes de transporte. El PID es entonces el campo que se emplea para distinguir los paquetes de transporte asociados a un determinado flujo elemental, de entre todos los demás.

De los 2^{13} posibles valores para el PID, han sido reservados por el estándar 17 para usos especiales. Es decir que existen 8175 posibles valores que pueden ser asignados a los diferentes flujos elementales, limitando entonces a este número la máxima capacidad de ESs que pueden ser contenidos en un TS.

Es responsabilidad del multiplexor, asegurar que cada flujo elemental tenga asignado un valor PID exclusivo de entre todos los posibles; por lo demás MPEG no condiciona la asignación de valores determinados a los distintos ESs.

En la tabla 2.2 se puede observar de forma general algunos de los valores PID reservados por MPEG, los cuales tienen asignaciones específicas contempladas por la norma.

Valor PID	Descripción
0x0000	Program Association Table (PAT)
0x0001	Condiciónal Access Table (CAT)
0x0002 – 0x000F	Reservado
0x0010 – 0x1FFE	Puede ser asignado como network_PID, program_map_PID, elementary_PID o para otros propósitos.
0x1FFF	Paquete nulo

Tabla 2.2 Tabla general de PIDs.

- transport_scrambling_control (control de enmascaramiento de transporte)

Este campo de dos bits indica el modo de cifrado de la carga útil del paquete del flujo de transporte. La cabecera del paquete TS y el campo de adaptación, cuando este último está presente, no deben ser cifrados. En el caso de un paquete nulo el valor de este campo deberá ser '00'.

La tabla 2.3 muestra la descripción de los posibles valores que puede tener el campo de control de enmascaramiento de transporte.

Valor	Descripción
00	No cifrado
01	Definido por el usuario
10	Definido por el usuario
11	Definido por el usuario

Tabla 2.3 Valores del control de enmascaramiento de transporte.

- adaptation_field_control (Control del campo de adaptación)

Este campo de dos bits indica si la cabecera del paquete TS está seguida por un campo de adaptación y/o una carga útil. En la tabla 2.4 se observan las diferentes combinaciones de estos bits y su descripción.

Valor	Descripción
00	Reservado para uso futuro de la ISO/IEC
01	Solo existe carga útil
10	Solo existe campo de adaptación
11	Campo de adaptación seguido de carga útil

Tabla 2.4 Valor del control del campo de adaptación.

- continuity counter (Contador de continuidad)

Este campo de 4 bits se incrementa con cada paquete TS de un mismo PID, es decir con cada paquete sucesivo perteneciente a mismo ES. Dicho contador permite al decodificador MPEG detectar la pérdida o ganancia de un paquete y de esta manera poder ocultar los errores que de otra forma podrían presentarse.

Este campo automáticamente cuenta desde '0' hasta su máximo valor. No deberá ser incrementado cuando el campo de control de adaptación sea igual a '00' o '10'.

2.1.3.2.1 *Campo de Adaptación del paquete MPEG-2*

Como la longitud de los paquetes PES en general no es múltiplo de 184 bytes, el último paquete MPEG-2 que sirve de transporte a un paquete PES deberá empezar por un campo de adaptación (*Adaptation Field, AF*), cuya longitud será el complemento a 184 del número de bytes que queden por transmitir para terminar este último paquete PES. [2]

Además de esta función de relleno, el campo de adaptación se utiliza también para la transmisión del Reloj de Referencia del Programa (*Program Clock Reference, PCR*), cuya cadencia de repetición mínima es de 10 veces por segundo, así como también para diversos tipos de datos opcionales.

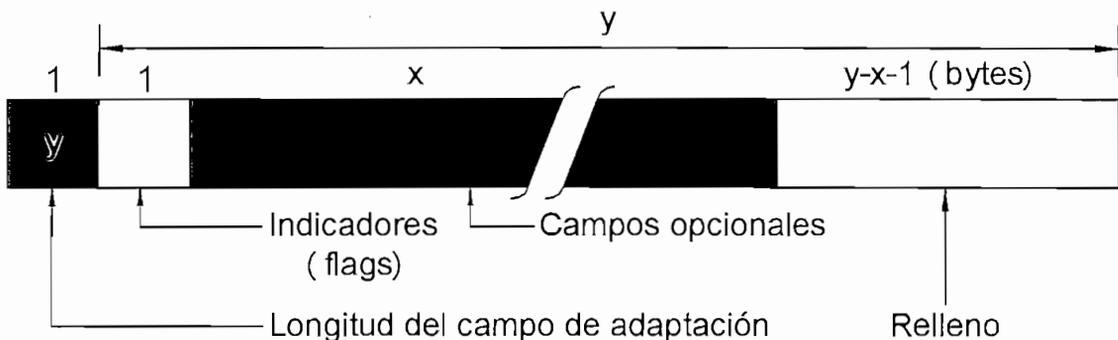


Figura 2.5 Detalle del campo de adaptación del paquete MPEG-2.

La figura 2.5 ilustra el formato general del campo de adaptación, cuyo contenido se especifica en la tabla 2.5.

Campo	Definición	No. de bits
Adaptation_field_length	Longitud total del campo en bytes, menos 1...(y)	8
Flags	Marcadores que indican la información transportada	8
Optional_fields	Campos opcionales de datos...(x bytes)	$x \times 8$
Stuffing	Campo de relleno...(y-1-x bytes)	$(y - 1 - x) \times 8$

Tabla 2.5 Constitución del campo de adaptación del paquete MPEG-2.

2.1.4 INFORMACIÓN ESPECÍFICA DE LOS PROGRAMAS (PSI)

Según se ha visto, un múltiplex de transporte MPEG-2 TS puede transportar varios flujos de información correspondientes por ejemplo a flujos de vídeo, audio o datos, cada uno compuesto por uno o varios PES. A fin de que el decodificador pueda orientarse dentro de este conjunto de paquetes, la norma MPEG define diferentes tipos de tablas de señalización que constituyen la información específica de los programas PSI (*Program Specific Information*) y de la red.

Esta PSI definida por MPEG para la Capa de Sistema (ISO/IEC 13818-1), comprende la inclusión dentro del flujo de transporte, de cuatro tipos de tablas, así:

- *Program Association Table (PAT)*
- *Conditional Access Table (CAT)*
- *Program Map Table (PMT)*
- *Private*

Es necesario indicar que, con el fin de que MPEG-2 se convierta efectivamente en el medio de multiplexaje y transporte para los datos DVB-S/RCS, la PSI propia de MPEG debe combinarse con las tablas de "Información de Servicio" o "Service Information (SI)" propias de los estándares DVB.

Cada tabla de información PSI o SI está constituida, según su importancia, por varias secciones (256 como máximo, con una longitud máxima de 1024 bytes excepto para ciertas tablas como las del tipo *Private* que pueden alcanzar los 4096 bytes). Dichas secciones están distribuidas a lo largo de una serie de paquetes de transporte identificados con un PID común. A continuación se describen brevemente las tablas que conforma la PSI.

2.1.4.1 Program Association Table (PAT)

La PAT es una tabla de inclusión obligatoria identificada con el valor PID de 0x0000H y contiene una lista completa de todos los programas o servicios disponibles en el TS.

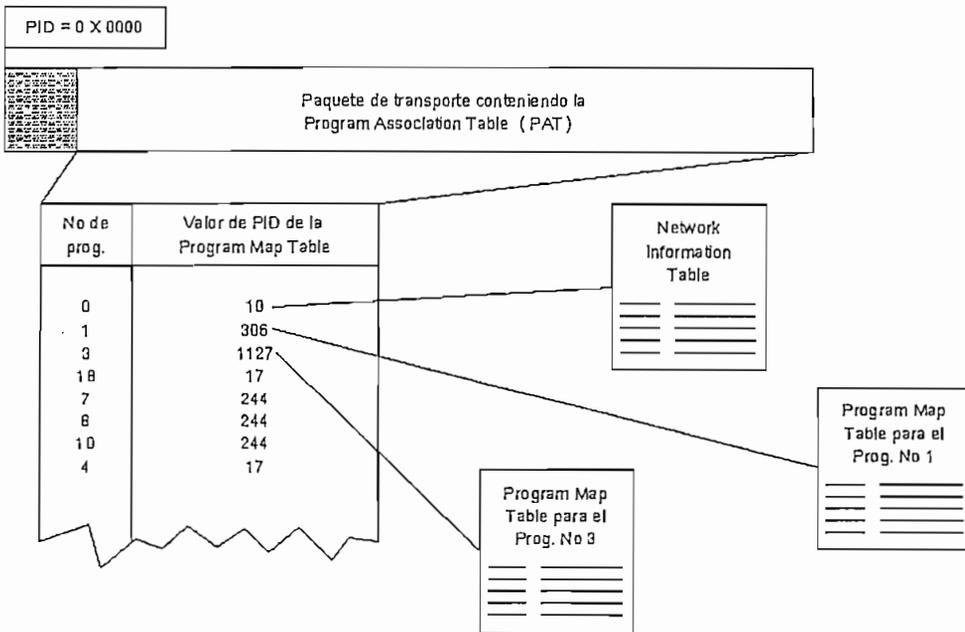


Figura 2.6 Forma de una Tabla PAT.

Cada programa aparece junto con el valor del PID de los paquetes que a su vez forman la tabla con la información que describe a dicho programa (*Program Map Table PMT*), tal como se muestra en la figura 2.6.

2.1.4.2 Program Map Table (PMT)

Existe una PMT presente por cada programa en el múltiplex. Es decir que cada servicio o programa que esté incluido en el TS tiene asociado a él una tabla PMT. Dicha tabla provee los detalles acerca del programa y de los flujos elementales que comprende.

La norma MPEG permite la asignación de valores de PIDs arbitrarios para la PAT, exceptuando obviamente los valores 0x0000 reservado para la PAT y 0x0001 reservado para la CAT. Sin embargo, las especificaciones DVB para las tablas SI restringen también el uso de los valores de PID que comprenden el rango desde 0x0002 hasta 0x001F.

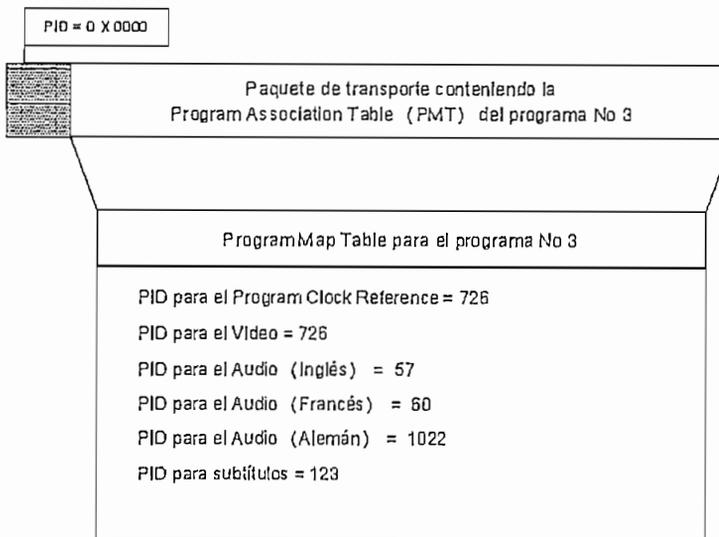


Figura 2.7 Estructura de una Tabla PMT.

La figura 2.7 muestra la constitución básica de una PMT. A través de la PMT, un decodificador podrá determinar que el ES codificado de vídeo correspondiente al programa No.3 se encuentra en los paquetes de transporte

identificados mediante el PID=726 y que el ES cuyos paquetes están etiquetados con PID=57 lleva el audio correspondiente en Inglés.

2.1.4.3 Conditional Access Table (CAT)

La CAT está presente siempre y cuando exista un programa de acceso condicional en el múltiplex. Utiliza el PID=0x0001 e indica los PID de los paquetes que transportan las informaciones necesarias para el descifrado de los programas de acceso condicional.

2.2 EL ESTÁNDAR DVB-S

El sistema europeo DVB-S es un conjunto de normas pensadas, en principio, para regular la transmisión de televisión digital. Es por eso que, inicialmente, a través de los sistemas DVB-S se ofrecieron servicios de difusión de canales de televisión digital, pero posteriormente, y como consecuencia de la evolución de los requerimientos del usuario, se contempló la posibilidad de soportar servicios multimedia y de acceso a Internet dentro del mismo sistema.

Es entonces que, en la actualidad las especificaciones DVB pueden utilizarse no sólo para la televisión propiamente dicha, sino también para la radiodifusión de una amplia gama de datos, vídeo, sonido, etc., acompañados de información auxiliar. Además, la necesidad de poner en marcha servicios interactivos a través de canales bidireccionales de comunicaciones por satélite, ha dado a lugar a la creación del estándar del sistema de canal de retorno DVB-RCS el cual será abordado posteriormente.

Vale la pena señalar que la tecnología DVB para satélite ha tenido una aceptación importante durante los últimos años, motivada no solo por la posibilidad de multiplexar diferentes servicios dentro de una misma portadora DVB-S, sino también por la fuerte reducción de precios de las estaciones remotas VSAT con receptores de este formato.

Por estos motivos es importante realizar una descripción de las normas DVB-S, ahondando en aquellos conceptos y recomendaciones que en capítulos posteriores serán de utilidad para construir la arquitectura y modelo de simulación de la "Red de Acceso a Internet", los cuales servirán para simular el comportamiento de la misma.

Se empezará entonces, describiendo las características físicas o de *hardware* del sistema DVB-S tales como interfaces, *transponders*, moduladores, etc., haciendo una breve revisión de los esquemas de codificación, modulación, corrección de errores, etc., adoptados por el mismo.

Se revisará con mayor detalle lo concerniente al Servicio de Información o *Service Information* (SI) propio de DVB-S, lo que permitirá entender cómo un decodificador puede orientarse dentro de la amplia gama de información presente en un flujo TS y como adquiere cierta información valiosa del sistema para interactuar dentro del mismo.

De la misma manera, se estudiarán las especificaciones DVB-S para la difusión de datos con el propósito de distinguir entre los distintos mecanismos de transporte de información que contempla DVB-S y los tipos de servicio que cada uno de ellos soporta. En esta sección se prestará importante atención al mecanismo de transporte MPE (*Multi-Protocol Encapsulation*), ya que como se verá más adelante, es precisamente este mecanismo el sugerido por la norma para el transporte de tráfico de Internet.

2.2.1 SISTEMA DE TRANSMISIÓN DVB-S [3]

2.2.1.1 Definición de sistema

El sistema está definido como aquel bloque funcional del equipo DVB-S que está encargado de la adaptación de las señales en banda base, entregadas por el múltiplex de transporte MPEG-2 a las características del canal satelital. El

siguiente procedimiento debe ser aplicado al flujo de datos MPEG-2 TS, antes de ser transmitido hacia el satélite:

- Adaptación del múltiplex de transporte y aleatorización para dispersión de energía
- Codificación externa (por ejemplo Reed-Solomon)
- Entrelazado de la codificación interna (códigos convolucionales)
- Codificación interna (por ejemplo código convolucional perforado)
- Filtrado de banda base para la modulación
- Modulación

La figura 2.8 muestra el diagrama de bloques del sistema de transmisión especificado por el estándar y cuya señal de entrada es el flujo de transporte MPEG-2 TS.

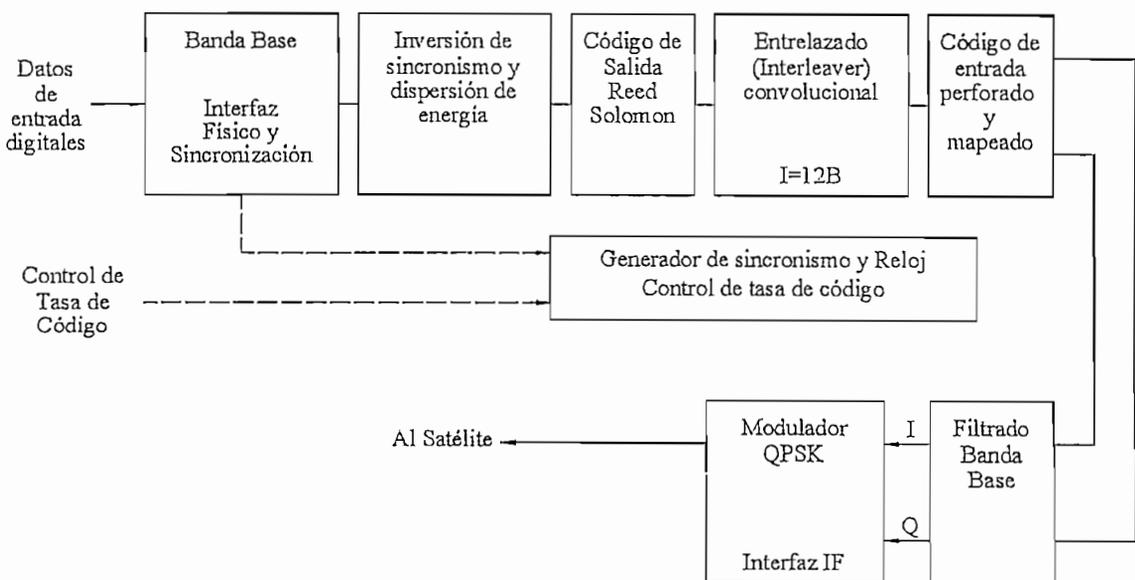


Figura 2.8 Diagrama de bloques funcional del Sistema de Transmisión DVB-S.

La norma DVB-S está orientada principalmente a soportar servicios DTH (*Direct To Home*), los cuales son afectados particularmente por las limitaciones de potencia del terminal; es por eso que la inmunidad al ruido y a la interferencia constituyen el principal objetivo de diseño del transmisor DVB-S, en lugar de una gran eficiencia espectral.

Para alcanzar niveles de potencia satisfactorios, sin afectar excesivamente a la eficiencia espectral, la norma DVB-S sugiere la utilización de una modulación QPSK, una concatenación de códigos convolucionales y Reed-Solomon. El código convolucional es capaz de ser configurado flexiblemente, permitiendo así la optimización en el desempeño del sistema para un determinado ancho de banda de un *transponder*.

El sistema es directamente compatible con la codificación de señales de televisión MPEG-2. La trama de transmisión del módem está sincronizada con los paquetes del múltiplex de transporte MPEG-2.

La técnica FEC (*Forward Error Correction*) adoptada en el sistema está diseñada para proveer un objetivo de calidad QEF (*Quasi Error Free*), cuando la señal recibida esté por debajo de los umbrales permitidos de S/N (relación señal a ruido) y S/I (relación señal a interferencia).

El QEF significa menos de un evento erróneo sin corregir por hora de transmisión, correspondiente a una BER de 10^{-10} a 10^{-11} en la entrada del demultiplexor MPEG-2.

2.2.1.2 Adaptación a las características del *transponder* satelital

Las transmisiones de los Servicios de Televisión Digital de Multiprogramas usarán satélites en ambas bandas de frecuencias FSS (*Fixed Satellite Service*) y BSS (*Broadcast Satellite Service*). La elección del ancho de banda del *transponder* está en función del satélite utilizado y la velocidad de datos requerida por el servicio.

2.2.1.3 Interfaces

El sistema DVB-S estará delimitado por los interfaces descritos en la tabla 2.6.

Localidad	Interfaz	Tipo de interfaz	Conexión
Estación transmisora	Entrada	Múltiplex de transporte MPEG-2	Desde el multiplexor MPEG-2
	Salida	70/140 MHz IF	Al equipo de RF
Estación receptora	Entrada	Múltiplex de transporte MPEG-2	Al multiplexor MPEG-2
	Salida	A ser definido	Desde el equipo de RF (<i>indoor unit</i>)

Tabla 2.6 Interfaces del sistema.

2.2.1.4 Codificación de canal

Una vez que se tiene el flujo de transporte MPEG-2, se debe realizar la codificación de canal. El proceso consiste, básicamente, en aplicar varios procesos de protección contra errores a la información digital antes de pasar a la modulación de RF (Radio Frecuencia).

Los procesos mencionados están indicados en el diagrama de bloques de la figura 2.8, y son detallados a continuación.

2.2.1.4.1 Inversión del sincronismo y dispersión de energía

Como se indicó anteriormente, la longitud total de un paquete dentro del múltiplex de transporte MPEG-2 TS es de 188 *bytes* e incluye un *byte* para la palabra de sincronismo (que puede ser por ejemplo 47H). El orden del proceso para la transmisión deberá comenzar siempre desde el MSB (*Most Significant Bit*) de la palabra de sincronismo.

Con el fin de cumplir con las Regulaciones de la ITU y para asegurar una adecuada transición binaria, la norma DVB-S señala que la información a la entrada del múltiplex debe ser aleatorizada.

La aleatorización se la consigue haciendo una operación lógica XOR entre la señal original y una secuencia binaria concreta. Este proceso resulta en una dispersión de energía, necesaria para no concentrarla demasiado en algunas partes de la secuencia binaria a transmitir. En recepción, al hacer el XOR entre la secuencia aleatoria obtenida y la secuencia binaria concreta utilizada en transmisión, se recupera la señal original.

Con el fin de proveer una señal de inicialización para la desaleatorización en el lado de recepción, el *byte* de sincronismo MPEG-2 del primer paquete de transporte en un grupo de ocho paquetes es invertido. Este proceso es conocido como "Adaptación del Múltiplex de transporte" o "Inversión de sincronismo".

2.2.1.4.2 Codificación Reed-Solomon

La señal aleatorizada en paquetes de 188 *bytes* es codificada con un código Reed-Solomon para generar un paquete protegido contra errores. Esta codificación es incluso aplicada al *byte* de sincronismo.

Este tipo de codificación introduce 16 *bytes* redundantes por cada paquete recibido, es decir, incrementa el número de *bytes* en una proporción de 204/188; permitiendo detectar y corregir en recepción hasta 8 *bytes* erróneos.

2.2.1.4.3 Entrelazado (*interleaving*)

El entrelazado, sirve para evitar ráfagas de errores consecutivas. Consiste básicamente, en dispersar a lo largo del tiempo las ráfagas de errores introducidas por el canal. Los 204 *bytes* del paquete se introducen sucesivamente en 12 registros (en el primer registro irán los *bytes* 1, 13, 25... en el segundo registro irán los bloques 2, 14, 26..., etc.).

Después se forma la nueva trama, concatenando los contenidos de los registros. Las ráfagas de errores del canal afectarán a *bytes* sucesivos de la nueva trama, pero al deshacer el entrelazado se repartirán a lo largo de la trama original. Así se aumenta la eficiencia de la decodificación Reed-Solomon, ya que al llegarle los errores más separados es más probable que los pueda corregir.

2.2.1.4.4 *Codificación convolucional y código perforado*

Es otra codificación para corrección de errores, la misma que emplea un código de Viterbi. Este tipo de codificación es muy adecuada para situaciones en las que la relación S/N es baja, y se adapta a diversos parámetros de la transmisión, introduciendo más o menos redundancia en función de las características particulares de cada enlace.

El código obtenido se denomina convolucional. Por ejemplo, en la situación más crítica se duplica el régimen binario de la señal (se aplica un código 1/2: 2 *bytes* de salida por cada uno de entrada), pero esto hace que la eficiencia espectral se reduzca a la mitad, ya que sólo la mitad de los *bytes* transmitidos contienen información.

Para situaciones mejores se pueden realizar perforaciones del código, es decir, reducir la redundancia enviando sólo parte de los *bytes* obtenidos a la salida del codificador convolucional: se obtienen relaciones 2/3, 3/4, 5/6 o 7/8, con menor protección de la señal.

2.2.1.4.5 *Filtrado banda base*

Se realiza un filtrado paso bajo de la señal para acotar su ancho de banda y evitar la interferencia entre símbolos. Es un filtrado de Nyquist con un filtro de coseno levantado de factor de *roll-off* del 35%, con lo que se tiene un ancho de banda a la salida del filtro de 1,35 veces el ancho de banda ideal de la señal en banda base.

Cabe mencionar que el factor de *roll-off* indicado por la norma es el fruto de estudios y simulaciones realizadas previamente a la aprobación final del estándar.

2.2.1.4.6 Modulación QPSK

La modulación es el proceso final del mecanismo de transmisión, necesario para poder transmitir la señal en canales concretos. Se trata de una modulación QPSK (*Quaternary Phase Shift Keying*) de amplitud constante, cuya información va incluida en la fase, muy robusta frente a ruidos atmosféricos. Además tiene una eficiencia espectral alta y ocupa un ancho de banda reducido.

Todas estas características la hacen especialmente adecuada para la transmisión de señal vía satélite, por su alta atenuación y bajo nivel de potencia en el receptor.

No está demás mencionar que recepción se realizan los pasos inversos que en transmisión con el fin de obtener la señal original.

2.2.2 INFORMACIÓN DEL SERVICIO (SI) [4]

Según se ha visto, el Múltiplex *MPEG-2 Transport Stream* puede contener varios programas o servicios, los cuales pueden ser recuperados completamente por el decodificador a través de los valores de los PIDs de los paquetes correspondientes gracias a la información adicional incluida en las tablas de la *PSI*.

Al igual que MPEG-2, el reporte técnico de la ETSI TR 101 211 v1.6.1, define un conjunto de tablas que conforma la denominada "Información del Servicio" o "*Service Information*" (SI), y que está presente con el propósito de proveer al terminal la información suficiente de la red para que éste entre a operar en el sistema. Cabe señalar que tanto la PSI como la SI trabajarán en conjunto dentro del mismo flujo de transporte, y que son los PIDs los que permitirán diferenciar unas tablas de otras.

La SI entonces incluye 4 tipos de tablas de inserción obligatoria dentro del flujo de transporte y 4 tipos de tablas opcionales.

Tablas Obligatorias:

- *Network Information Table (NIT)*
- *Service Description Table (SDT)*
- *Event Information Table (EIT)*
- *Time & Date Table (TDT)*

Tablas Opcionales:

- *Bouquet Association Table (BAT)*
- *Running Status Table (RST)*
- *Time Offset Table (TOT)*
- *Stuffing Tables (ST)*

Cada tabla está constituida, según su importancia, por una o varias secciones (256 como máximo, con una longitud máxima de 1.024 bytes excepto para tablas tipo "Private" y "EIT" que pueden alcanzar los 4.096 bytes).

Dichas secciones están distribuidas a lo largo de una serie de paquetes de transporte identificados con un PID común. A continuación se detalla cada tabla que conforma la SI.

2.2.2.1 *Network Information Table (NIT)*

Esta tabla constituye por definición el programa '0x0000' del múltiplex de transporte MPEG-2. Se transporta por los paquetes identificados con PID=0x0010.

La NIT provee un grupo de flujos de transporte y la información de ajuste o *set-up* acerca de la red física usada para transmitir el TS, como por ejemplo:

frecuencias del canal, detalles del *transponder* del satélite, características de modulación, detalles de redes alternativas disponibles, etc.

Esta tabla podría ser empleada durante el proceso de estructuración de la IRD y la información de ajuste usada para este fin, puede ser grabada en una memoria no volátil.

Las siguientes reglas aplican a la NIT:

- a) La transmisión de la NIT es mandatoria para el sistema actual de entrega de servicios.
- b) La NIT que describe el sistema actual de entrega de servicios es válida si y solo si contiene descriptores aplicables a dicho sistema. Esta regla describe las condiciones bajo las cuales una NIT contiene información válida. Si un IRD satelital recibe un descriptor de un sistema de entrega satelital para el sistema de entrega actual, entonces éste es válido, o si un IRD terrestre recibe un descriptor de un sistema de entrega terrestre para el sistema de entrega actual, también es válido. Pero si un IRD terrestre recibe un descriptor de un sistema de entrega satelital para el sistema de entrega actual, éste se asume como inválido por el IRD.
- c) Si una NIT válida para el sistema de entrega actual está presente en el flujo de bits SI, ésta enlistará todos los TSs del sistema de entrega actual.
- d) El flujo SI tendrá al menos 8 paquetes TS por cada 10 segundos llevando datos NIT o paquetes nulos.

El flujo SI usa dos etiquetas relacionadas al concepto de sistema de entrega, nombradas *network_id* y *original_network_id*. Esta última es propuesta para soportar la identificación única de un servicio contenido en un TS, aún si éste

ha sido transferido a otro sistema de entrega diferente al sistema de entrega donde se originó. Un TS puede ser referenciado únicamente a través de la ruta *original_network_id/transport_stream_id*.

Un servicio se puede referenciar únicamente a través de la ruta *original_network_id/transport_stream_id/service_id*. Por lo tanto, la etiqueta *network_id* no es parte de esta ruta.

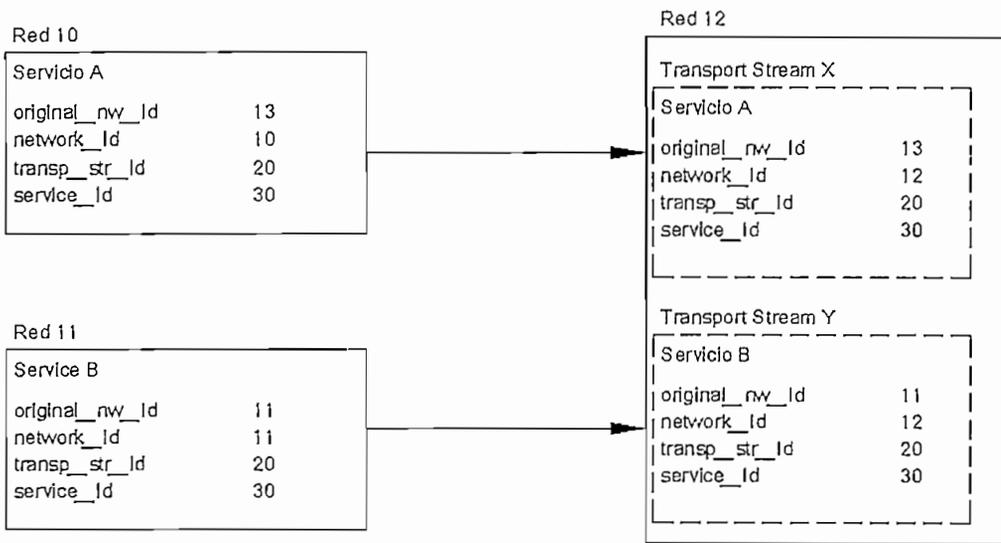


Figura 2.9 Ejemplo de Transferencia de Servicios a un nuevo Sistema de Entrega.

Además, cada *service_id* será único dentro de cada *original_network_id*. Cuando un servicio (contenido dentro de un TS) es transferido a otro sistema de entrega, solamente la etiqueta *network_id* cambia, mientras que la etiqueta *original_network_id* se mantiene inalterada, tal y como lo muestra la figura 2.9.

2.2.2.2 Service Description Table (SDT)

La SDT es usada para listar los nombres y otros parámetros de los servicios dentro de los TSs. Para cada TS existirá una subtabla SDT por separado. Las siguientes reglas aplican a la SDT para mejorar la adquisición de servicios:

- a) La transmisión de la SDT para el TS actual es obligatoria.
- b) El flujo de bits SI listará en la SDT, de al menos un TS particular, todos los servicios de ese TS.
- c) Cualquier SDT para otro diferente TS que el actual, listará todos los servicios de ese TS.
- d) Es altamente recomendable que los *service_ids*, una vez asignados a un servicio específico dentro de una red, se mantengan inalterables para permitir a los IRDs implementar características tales como listas de canales favoritos, etc.

La SDT se transporta por paquetes identificados con PID=0x0011.

2.2.2.3 *Event Information Table (EIT)*

Se usa para transmitir información relativa a los acontecimientos en curso o futuros en el múltiplex MPEG-2 TS recibido, y eventualmente sobre otros múltiplex MPEG, tal como: denominación, hora de comienzo, duración, etc. Se transporta por los paquetes identificados con PID=0x0012.

2.2.2.4 *Time and Date Table (TDT)*

La TDT transmite el tiempo UTC (*Universal Time Coordinated*) actual codificado como MJD (*Modified Julian Date*). Puede ser usada para sincronizar el reloj interno de una IRD.

La TDT, según la norma, deberá ser transportada por paquetes identificados mediante PID=0x0014.

2.2.2.5 *Bouquet Association Table (BAT)*

La BAT provee un grupo de servicios que sirven como base sobre la cual una IRD presentará los servicios disponibles para un usuario. La transmisión de esta tabla es opcional. La siguiente regla aplica a la BAT para mejorar la consistencia en el flujo de bits SI y simplificar el procesamiento en las IRDs.

El flujo de bits SI enlistará en cada sub-tabla BAT todos los servicios pertenecientes a ese *bouquet* o conjunto de servicios.

Es de señalar que un servicio puede pertenecer a más de un *bouquet*. Esta regla crea consistencia a través de diferentes TSs los cuales son accesibles a la IRD.

La tabla BAT es transportada por paquetes identificados con PID=0x0011.

2.2.2.6 *Running Status Table (RST)*

Las tablas RST son usadas para actualizar rápidamente el estado de uno o más eventos que están ocurriendo. Éstas son enviadas una sola vez, en el momento que el estado de un evento cambia, al contrario de otras tablas SI las cuales normalmente son transmitidas repetidamente. Es por eso que no existe ningún mecanismo de actualización para las RSTs.

Las tablas RSTs se transportan por paquetes identificados con PID=0x0013.

2.2.2.7 *Time Offset Table (TOT)*

La TOT transmite el tiempo UTC actual incluido dentro de la información de desplazamiento codificado como MJD. Puede ser usada para sincronizar el reloj interno de una IRD. La transmisión de la TOT es opcional, pero si está presente será transmitida al menos cada 30 segundos.

Se transportan por paquetes identificados con PID=0x0014, tal como las TDT.

2.2.2.8 *Stuffing Table (ST)*

Estas tablas de "relleno" se emplean para invalidar tablas que ya no sirven. Por ello usan paquetes que comparten valores de PID con otros tipos de tablas, tales como: 0x0010; 0x0011; 0x0012 0x0013 y 0x0014.

2.2.2.9 **Códigos PID especificados por DVB-SI**

La tabla 2.7 resume los valores de PID que deben emplearse en los paquetes del múltiplex tipo "Transport Stream" que llevan secciones con la "Información del Servicio (SI)", de acuerdo con las especificaciones para DVB.

Tabla	Valor de PID
PAT	0X0000
CAT	0X0001
TSDT	0X0002
Reservados	0x0003 a 0x000F
NIT,ST	0X0010
SDT,BAT, ST	0X0011
EIT, ST	0X0012
RST,ST	0X0013
TDT,TOT, ST	0X0014
Sincronización de Red	0x0015
Reservados para usos futuros	0x0016 a 0x001D
DIT*	0X001E
SIT*	0X001F

Tabla 2.7 Valores de PID para las Tablas SI.

* Las tablas de tipo "Discontinuity Information Table (DIT)" y "Selection Information Table (SIT)" se emplean solamente en flujos de transporte grabados, y por lo tanto parciales. Dichas tablas transportan la información requerida para describirlos.

2.2.2.10 Arquitectura de las Tablas de Información del Servicio (SI)

Las tablas PSI definidas por MPEG-2 y las tablas SI especificadas por DVB están constituidas por una o más secciones que se distribuyen a lo largo de una serie de paquetes de transporte del múltiplex.

Una sección es la estructura sintáctica que debe emplearse para insertar las citadas tablas en los paquetes de transporte. La estructura sintáctica usada en DVB para las tablas SI, coincide con la especificada por MPEG-2 para las secciones de las tablas PSI de tipo "Private".

Las secciones pueden tener longitud variable. Dentro de cada tabla SI o PSI, la longitud de las secciones está limitada a 1K (1.024 bytes), excepto en el caso de las secciones de las tablas "Private" y "EIT" cuya longitud está limitada a 4Kbytes (4.096 bytes) por las normas. [5]

2.2.2.10.1 Distribución de las secciones en los paquetes de transporte

Al contrario que la distribución de los PES en los paquetes TS, las secciones de las tablas PSI y SI no tienen que empezar ni acabar forzosamente con un paquete de transporte. Sin embargo, si un paquete TS contiene el comienzo de al menos una sección, su "indicador de carga útil" debe ponerse en '1'.

El comienzo de una nueva sección en un paquete TS se indica por el campo "*pointer_field*", que debe ser el primer byte de la carga útil del citado paquete y proporciona el desplazamiento (*offset*), en bytes, desde el final del "*pointer_field*" hasta el primer byte de dicha sección.

Así, un valor de 0x0000 indica que la sección comienza inmediatamente después del "*pointer_field*". Si ninguna sección tiene su comienzo dentro del paquete TS, no habrá "*pointer_field*" y el "indicador de inicio de carga útil" se pondrá a '0'.

Solamente hay un "*pointer_field*", puesto que el comienzo de cualquier otra sección de una tabla se puede identificar fácilmente contando la longitud de las secciones precedentes, ya que la sintaxis no permite que haya huecos entre ellas. La figura 2.10 ilustra esta situación en el caso general.

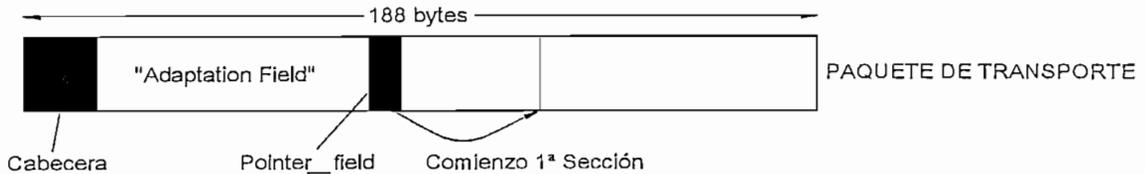


Figura 2.10 Primer comienzo de sección dentro de un paquete TS.

Si una sección termina antes del final de un paquete de transporte, pero no es conveniente abrir otra dentro del mismo paquete, se debe emplear un mecanismo de relleno que emplee para cada byte sobrante el valor 0xFF. Consecuentemente, este valor no debe usarse para el byte que identifica el tipo de tabla: "*table_id*".

Sin embargo, también puede usarse como método de relleno de los paquetes de transporte el mecanismo que usa el "*adaptation_field*" y que se explicó en la sección 2.1.3.2.1.

2.2.3 ESPECIFICACIONES PARA LA DIFUSIÓN DE DATOS [5]

Hasta el momento se ha definido la manera de transportar un programa, por ejemplo de televisión, sobre un flujo de información MPEG-2 TS. Queda en este punto definir los métodos para transportar datos dentro del TS para construir redes de comunicación VSAT sobre la tecnología DVB-S.

Entiéndase como transporte de datos a las descargas de *software* desde un satélite, entrega de los servicios de Internet sobre canales de difusión (*IP tunnelling*), televisión interactiva, etc.

Los estándares DVB-S EN 301 192 v1.3.1 y ETSI TR 101 202, proveen los recursos necesarios para la entrega de flujos MPEG-2 TS a través de una variedad de medios lógicos de transmisión. Son precisamente estos medios lógicos o mecanismos de transporte de datos los que serán revisados en esta sección, lo que nos permitirá escoger el método más apropiado para desarrollar el modelo y la arquitectura de la “Red VSAT de Acceso a Internet”.

A continuación se detallarán entonces, dichos métodos de transporte de datos priorizando el análisis del Encapsulado MPE por ser aquel que la norma sugiere para el transporte de tráfico de Internet y por consiguiente el de mayor relevancia para el desarrollo del presente proyecto. Así, se tienen los siguientes métodos:

- Tubo de Datos (*Data Pipe*)
- Flujo Continuo de Datos (*Data Streaming*)
- Carrusel de Datos (*Data Carrousel*)
- Carrusel de Objetos (*Object Carrousel*)
- Encapsulamiento Multiprotocolo MPE

2.2.3.1 Tubo de datos (*Data pipe*) [6]

Tubo de Datos o *Data Pipe*, es un mecanismo simple de transporte de datos extremo a extremo (E2E). No existe una relación de tiempo entre los datos y otros tipos de PES. Este mecanismo necesita de un procedimiento explícito de segmentación y reensamblado de los datos mediante un nivel de adaptación fuera de la especificación DVB-S, razón por la cual no se contempla para ser el medio de transporte para la red propuesta en este proyecto.

2.2.3.2 Carrusel de datos (*Data carrousel*)

Consiste en la transmisión periódica de archivos de datos. El conjunto de datos puede ser de cualquier formato o tipo, sin embargo no es recomendable para el tráfico de Internet. Un ejemplo de uso son las guías electrónicas de

programas (EPGs). Los datos se envían usando secciones de tamaño fijo DSMCC.

2.2.3.3 Flujo continuo de datos (*Data streaming*)

Incluye la transmisión sincronizada o asincrónica de flujos continuos de datos E2E. Los datos se transportan en flujos de paquetes elementales PES por lo que las funciones de segmentación y reensamblado son automáticas. Sin embargo de esto último, la norma no recomienda este mecanismo para la transportación de tráfico de Internet, por lo que este método tampoco se toma en cuenta para la propuesta de red.

2.2.3.4 Carrusel de objetos (*Object carousel*)

Facilita la transmisión de un grupo estructurado de objetos desde un servidor de difusión hacia un conjunto de receptores (clientes) usando un directorio de objetos, archivo de objetos o un flujo de objetos. El directorio de objetos se utiliza por el receptor para encontrar la posición de los archivos dentro del flujo de objetos, y como los otros tres métodos no se recomienda para el tráfico de Internet.

2.2.3.5 Encapsulamiento Multi-Protocolo (MPE)

El método de encapsulamiento MPE incluye un mecanismo de transporte para una gran variedad de servicios de comunicación de datos. Esta técnica se basa en las secciones DSM-CC¹ (*Digital Storage Media – Command and Control*) y está orientada a proveer un servicio de emulación de LAN para el intercambio de datos. [6]

Esta técnica es la recomendada para la transmisión de datos de Internet y al igual que en el caso del mecanismo de transporte *Data Streaming*, las

¹ DSM-CC es una técnica mediante la cual se fragmenta los datagramas IP en secciones o mensajes para su posterior encapsulación en paquetes MPEG-2.[1]

funciones de segmentación y reensamblado son automáticas; éstas son razones suficientes para ser escogido como el mecanismo que permitirá encapsular datagramas dentro del flujo de transporte MPEG-2 TS de la "Red VSAT de Acceso a Internet".

Debido a la naturaleza de difusión de las redes DVB, la seguridad de los datos es de mucha importancia para el sistema, por lo que esta técnica de encapsulación, además de las ventajas anteriormente señaladas, permite una transmisión segura de datos a través de la encriptación de los paquetes y del cambio dinámico de direcciones MAC, aspectos que se revisarán en detalle más adelante.

2.2.3.5.1 Transporte de datos con MPE

Al igual que DSM-CC, el método de encapsulación MPE consiste básicamente en una técnica mediante la cual se toman los datagramas provenientes de la capa Internet (datagramas IP), se los fragmenta en Secciones MPE y posteriormente se las encapsula dentro de los paquetes de transporte MPEG-2 TS.

Los datagramas entonces se transportan en "Secciones de Datagramas MPE", las cuales están sujetas al formato de sección DSM-CC para "Datos Privados", con ciertas variantes propias del encapsulado MPE y que serán apreciadas en el transcurso de este tema.

El formato que se adopta para la sección MPE resulta eficiente para organizar los datagramas en los paquetes de flujo de transporte MPEG2, y además para soportar un "filtrado de datagramas" basado en la dirección MAC usando demultiplexores existentes en el equipo receptor, ya sean éstos en *hardware* o en *software*.

La sintaxis del formato de la cabecera incluido el campo CRC (un total de 16 *bytes* de sobrecarga) de las Secciones de Datagrama MPE están definidas en la tabla 2.8.

Sintaxis	No. De bits
table_id	8
Section_syntax_indicator	1
private_indicator	1
Reserved	2
Section_length	12
MAC_address_6	8
MAC_address_5	8
Reserved	2
Payload_scrambling_control	2
Address_scrambling_control	2
LLC_SNAP_flag	1
current_next_indicator	1
Section_number	8
Last_section_number	8
MAC_address_4	8
MAC_address_3	8
MAC_address_2	8
MAC_address_1	8
IP_datagram_data_byte	8
stuffing_byte	8
Checksum	32
CRC_32	32

Tabla 2.8 Sintaxis de la Sección de Datagrama MPE.

El formato de la sección permite también fragmentar datagramas en múltiples secciones. Si la longitud de un datagrama es menor o igual a 4080 *bytes* (incluida la posible cabecera LLC/SNAP), el datagrama será enviado en una sola sección.

En el caso de IP y de LLC/SNAP las cabeceras serán omitidas, el MTU (*Maximun Transfer Unit*) se configura en 4080 *bytes* o menos, de esta manera los datagramas nunca serán fragmentados.

En el caso de que la cabecera IP y LLC/SNAP esté presente el MTU se configura en 4074 o menos para que no exista fragmentación.

En las secciones MPE, los 6 *bytes* de la dirección MAC han sido divididos en dos grupos dentro de la cabecera MPE, y están distribuidos de la forma en que indica la figura 2.11. Este cambio dinámico de la dirección MAC le permite al receptor filtrar más rápidamente las secciones ya que generalmente los dos primeros bytes de la dirección MAC presentes en la cabecera MPE son filtrados por *hardware* y son los *bytes* que probablemente más diferencien un receptor de otro.

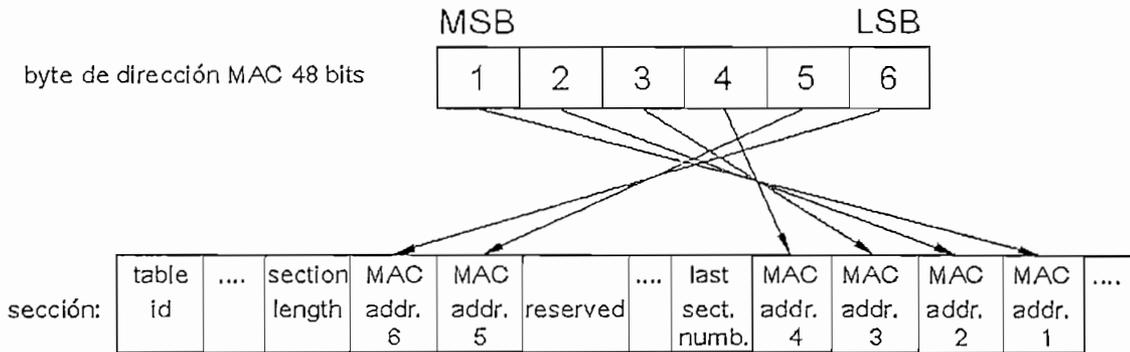


Figura 2.11 Organización de la dirección MAC en campos de sección MPE. [6]

Como ya se mencionó, algunos demultiplexores son capaces de filtrar los *bytes* 5 y 6 mediante *hardware*, mientras que los *bytes* 1,2,3 y 4 los filtran a través de software.

Es recomendable entonces que los dos *bytes* de la dirección MAC, los cuales probablemente diferencien más a los receptores, sean puestos en los *bytes* 5 y 6 de la cabecera MPE.

Esta situación descrita es muy normal en las direcciones MAC IEEE, por lo que es recomendable que todas las direcciones MAC sean construidas de esta manera en las Secciones MPE.

Como se mencionó anteriormente una característica importante de MPE es su mecanismo de protección de la señal; es así, que dentro de su cabecera el campo denominado *payload_scrambling_control* indica la aleatorización de la carga útil, con lo que se puede lograr cierta inmunidad al monitoreo de flujo de datos privados.

Existe también la posibilidad de aleatorizar la dirección MAC a través del campo *address_scrambling_control*, lo cual provee mayor seguridad gracias al cambio dinámico de la misma. Con el cambio de la palabra de control que se usa para aleatorizar la dirección MAC periódicamente, el monitoreo del flujo de datos puede ser prevenido, ya que mediante este método no se puede determinar el destino de un datagrama en particular observando las direcciones MAC de la sección. Esto fortalece la seguridad ya que la recolección de los datagramas destinados a un determinado receptor es difícil.

La sección MPE podría contener relleno después del datagrama. Los *bytes* de relleno se pueden utilizar, por ejemplo, para que la carga útil de la sección sea múltiplo de un tamaño de bloque determinado, situación que es común cuando se usa un código de encriptación de bloque. El valor de estos *bytes* no se especifica y en el caso de la encriptación de la carga útil, el valor de ellos no debería ser fijo, puesto que esto podría ayudar a romper la encriptación.

Finalmente, la sección de datagrama tiene un campo de *checksum* o uno de CRC_32 al final, dependiendo del valor del campo *section_syntax_indicator*. Se recomienda utilizar CRC_32 puesto que éste provee una mejor protección en contra de los errores de *bits*, ya que se puede chequear por *hardware* en la mayoría de los demultiplexores (tipo *hardware*), mientras que el *checksum* se tiene que chequear por *software* normalmente.

2.2.3.5.2 Información en la SI

El servicio de difusión de datos en DVB, deberá indicar la transmisión de datagramas a través de la inclusión de uno o más descriptores de difusión de datos dentro de la SI.

Para los servicios que utilizan encapsulación multiprotocolo, el descriptor *data_broadcast_descriptor* estará presente en la tabla SDT o en la EIT.

Cada descriptor se deberá asociar a un flujo a través de un identificador denominado *component_tag*. En particular, el valor del campo *component_tag* deberá ser idéntico al valor del campo *component_tag* del descriptor *stream_identifier_descriptor* que puede estar presente en la tabla PMT (ver sección 2.1.4.2) del flujo que se usa para transmitir los datagramas.

2.3 EL ESTÁNDAR DVB-RCS [7]

Las redes satelitales basadas en la tecnología DVB-S han tenido un éxito considerable especialmente para los servicios fundamentalmente de difusión que requieren poca o ninguna interacción por parte del usuario (por ejemplo difusión digital de televisión y radio).

Las necesidades del usuario, generadas principalmente por el desarrollo de Internet, motivó que estos servicios unidireccionales requieran un nivel de interactividad y un primer enfoque para conseguirlo fue la utilización de canales de retorno terrestre sobre la RTC o la RDSI [8], tal y como se explicó en la sección 1.2 del capítulo anterior.

En principio, un canal de retorno terrestre sobre la RTC de baja velocidad no representaba un mayor problema para cierto tipo de servicios interactivos, donde la asimetría de comunicación entre lo que el usuario envía y lo que recibe es considerable (por ejemplo la navegación Web).

Sin embargo el canal de retorno terrestre no permite al operador satelital tener control sobre la calidad de servicio o QoS de dicho canal, ya que generalmente éste se encuentra bajo la administración de otro operador de red y además su utilización puede no ser la óptima, ya que la asignación de recursos es habitualmente exclusiva mientras dure la conexión.

Un enlace de retorno satelital puede habilitar el acceso múltiple (por ejemplo MF-TDMA) de tal forma que la ocupación de los recursos puede optimizarse en función de la demanda de transmisión de los clientes.

Sumado a estos inconvenientes, tal como se mencionó en el capítulo anterior, un canal de retorno terrestre limita la ubicuidad de servicio dentro de la zona de cobertura de los sistemas satelitales, es decir, para prestar servicio en una determinada localidad no solamente se requiere que ésta se encuentre dentro del área de cobertura del satélite, sino que además exista una infraestructura de comunicación terrestre habilitada.

Este problema es de especial relevancia tomando en cuenta que es precisamente el mercado que se encuentra en zonas donde no existe una infraestructura terrestre, aquel de mayor importancia para las comunicaciones.

Una comunicación bidireccional por satélite permite resolver los inconvenientes generados por la dependencia a una infraestructura de comunicación terrestre y además mantiene las ventajas inherentes en este tipo de comunicaciones.

Las comunicaciones bidireccionales por satélite a través de las pequeñas estaciones VSAT han sido una realidad desde hace varios años pero no han estado enfocadas al mercado del usuario común y constituyen soluciones propietarias costosas con los consiguientes problemas de compatibilidad y dependencia hacia un único proveedor de servicios.

Es entonces, que de la necesidad de una comunicación bidireccional por satélite asequible al usuario común, surge a principios del año 2000 el estándar de la ETSI EN 301 790 v1.4.1 para el canal de retorno satelital DVB-RCS el cual es una norma que permite la construcción de sistemas multimedia completamente interactivos por satélite con la gran ventaja de que ésta se basa en la tecnología DVB cuya aceptación respecto al canal de difusión DVB-S ha sido amplia.

La norma DVB-RCS tiene claramente una orientación a cubrir diferentes tipos de mercado, que van desde los usuarios domésticos, profesionales y corporativos de allí que las estaciones interactivas cuentan con diferente configuración y capacidades de acuerdo al escenario de su utilización.

De esta forma el estándar DVB-RCS no solo busca eliminar la dependencia a las redes terrestres sino también incrementar las velocidades de transmisión de las estaciones remotas y así ampliar el rango de servicios interactivos que se ofrecen mediante estos sistemas bidireccionales.

Cabe indicar que el estándar DVB-RCS presenta la especificación para la provisión del canal de retorno para redes satelitales interactivas a través de Satélites Geoestacionarios (GEO) con Terminales Satelitales del Canal de Retorno fijos o RCST, pudiéndose aplicar dicha especificación a todas las bandas de frecuencia asignadas a los servicios satelitales GEO.

2.3.1 MODELOS DE REFERENCIA PARA REDES SATELITALES INTERACTIVAS DVB

2.3.1.1 Modelo de pila de protocolos

Para servicios interactivos destinados a soportar difusión hacia el usuario final con un canal de retorno satelital, un simple modelo de comunicaciones consiste en las siguientes capas o niveles:

- **Capa física:** donde todos los parámetros físicos para la transmisión son definidos.
- **Capa transporte:** define todas las estructuras relevantes de datos y protocolos de comunicación como contenedores de datos.
- **Capa aplicación:** es el *software* de aplicación que interactúa con el usuario (por ejemplo aplicaciones de compras desde el hogar, etc.).

Un modelo simplificado de las capas de OSI fue adoptado por la norma DVB-RCS para facilitar la producción de especificaciones para estas capas. La figura 2.12 muestra las capas más bajas de este modelo simplificado e identifica algunos de los parámetros importantes para las dos capas más bajas.

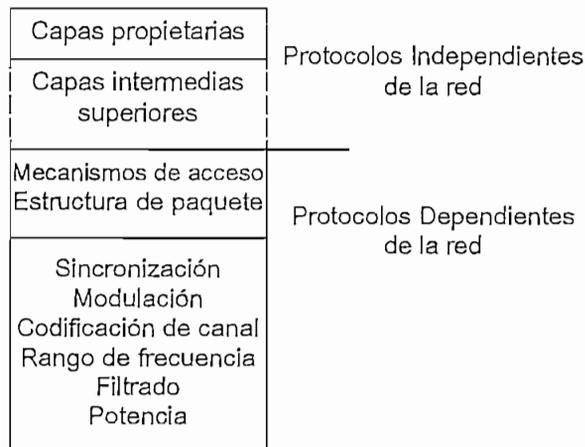


Figura 2.12 Estructura en capas de un Modelo de Referencia para un Sistema Genérico DVB-S/RCS.

Cabe señalar que, como se puede apreciar en la figura 2.12, el estándar únicamente especifica los aspectos dependientes de la Red Satelital Interactiva, es decir contempla exclusivamente los Protocolos Dependientes de la Red.

2.3.1.2 Modelo Genérico del Sistema

La figura 2.13 muestra el Modelo de Referencia del Sistema Genérico que se utilizará dentro de DVB como base para la provisión de servicios interactivos a las RCSTs.

Como se puede apreciar en la figura anterior, en el Modelo de Referencia del Sistema se establecen dos canales entre el proveedor de servicios y el usuario RCST, los cuales son:

- Broadcast Channel o Canal de Difusión

Un canal de difusión unidireccional que incluye vídeo, audio y datos, se establece desde el proveedor de servicios hasta los usuarios. Éste puede incluir el *Forward Interaction Path* (Camino Interactivo de Ida).

- Interaction Channel o Canal Interactivo

Un canal interactivo bidireccional se establece entre el proveedor de servicios y usuarios o entre usuarios para propósitos interactivos. El canal interactivo está formado por:

- *Return Interaction Path (Canal de Retorno)*: se establece desde el usuario al proveedor de servicios. Se usa para hacer demandas al proveedor de servicios o al usuario según sea el caso, para contestar preguntas o transferir datos.
- *Forward Interaction Path (Canal de Ida Interactivo)*: se establece desde el proveedor de servicios al usuario. Se usa para proporcionar la información desde el proveedor de servicios a los usuarios y cualquier otra comunicación que se requiere para la provisión de servicios interactivos. Puede ser parte del *Broadcast Channel* o Canal de Difusión. Es posible que este canal no se requiera en algunas

aplicaciones simples que hacen uso del *Broadcast Channel* para el envío de datos hacia el usuario.

La RCST, por su parte, está formada por la *Network Interface Unit* (consta del Módulo de Interfaz de Difusión y el Módulo de Interfaz Interactivo) y la *Set Top Unit*, tal como se aprecia en la figura 2.13. La RCST provee el interfaz para ambos canales, el difusivo y el interactivo. El interfaz entre la RCST y la red interactiva es definido por el Módulo de Interfaz Interactivo.

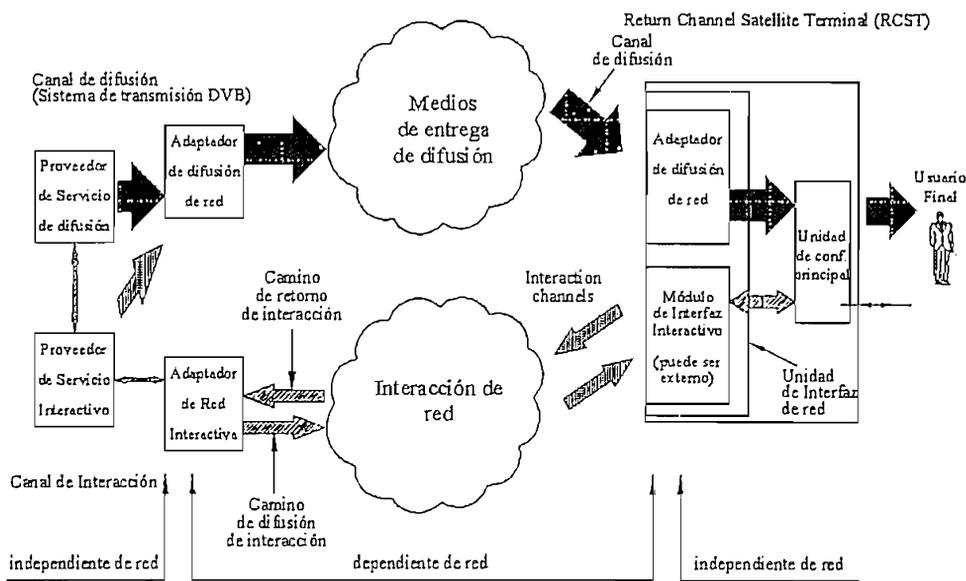


Figura 2.13 Modelo de Referencia de un Sistema Genérico para Sistemas Interactivos.

2.3.1.3 Modelo de referencia para Redes Satelitales Interactivas

Una Red Satelital Interactiva Global, dentro de la cual un gran número de RCSTs se encuentran operando, deberá presentar dentro de su estructura los siguientes bloques funcionales, tal como se ilustra en la figura 2.14.

- Network Control Center (NCC)

Un NCC o Centro de Control de RED, proporciona Control (*Control*) y Funciones de Monitoreo (*Monitoring Functions*) (CMF) a toda la red. El NCC

genera entonces, señales de control y de sincronización para el correcto funcionamiento de la Red Satelital Interactiva, las mismas que serán transmitidas por una o varias Estaciones Alimentadoras DVB-S o *Feeder Stations*.

- Traffic Gateway (TG)

Un TG o Pasarela de Tráfico, recibe las señales de retorno de las RCSTs. Provee funciones de contabilidad, servicios interactivos y/o conexiones a proveedores de servicios propietarios y privados (bases de datos, *pay-per-view*, descarga de *software*, *tele-shopping*, *tele-banking*, servicios financieros, acceso a bolsas de valores, juegos interactivos, etc.) y a redes (Internet, ISDN, PSTN, etc.).

- Feeder (Alimentador)

Un Alimentador transmite la señal del Enlace de Ida (*Forward Link*), la cual consiste de un *uplink* estándar DVB-S, sobre el cual se multiplexan datos de usuario y/o señales de control y sincronización necesarias para el funcionamiento de la Red Satelital Interactiva.

En la figura 2.14 se puede observar un enlace de difusión DVB-S, cuya función es la de transportar la señalización desde el NCC y el tráfico de usuario hacia las RCSTs, y que trabaja bajo el estándar DVB-S. Dicha señalización, la cual es necesaria para operar el sistema de enlace de retorno, será llamada "*Forward Link Signalling*" (FLS).

Cabe mencionar que tanto el tráfico de usuario y el *Forward Link Signalling* pueden ser llevados sobre diferentes enlaces de *Forward Link*. De hecho, son posibles varias configuraciones de RCST dependiendo del número de receptores DVB-S que estén presentes en la RCST.

- Sincronización de la RCST
- Formato de Ráfaga
- Aleatorización para la Dispersión de Energía
- Codificación de Canal
- Modulación

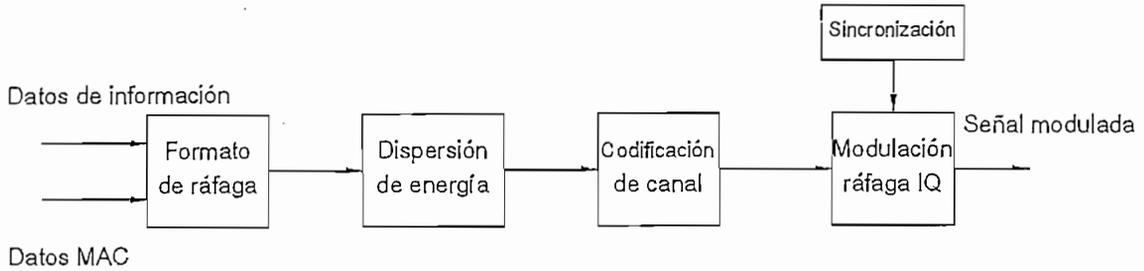


Figura 2.15 Diagrama de Bloques del procesamiento de señal en banda base del canal de retorno en un transmisor RCST.

Cada uno de los Bloque Funcionales mencionados anteriormente, será descrito en detalle a continuación.

2.3.3.1 Sincronización de la RCST

2.3.3.1.1 Control de temporización

La sincronización de la RCST es una característica importante dentro de las redes satelitales interactivas y se especifica por la norma como el primero de los procesos previos a la transmisión de información. Es así que el estándar impone limitaciones en las RCSTs para obtener una máxima eficiencia en el sistema TDMA con una mínima interferencia entre usuarios y un máximo *throughput*.

El esquema de sincronización del sistema, se basa en cierta información esencial de temporización contenida dentro del FLS, la cual es:

- El Reloj de Referencia de Red o NCR (*Network Clock Reference*).
- Señalización presente en las secciones privadas DVB/MPEG2-TS.

El NCR es distribuido con un PID específico dentro del flujo de transporte MPEG-2 que lleva el FLS. La distribución del NCR sigue el mismo mecanismo de distribución del Reloj de Referencia de Programa PCR (*Program Clock Reference*) descrito en el estándar ISO/IEC 13818-1, el cual normalmente se deriva de un codificador de vídeo MPEG, pero para DVB-RCS el NCR se deriva desde el Reloj de Referencia del NCC, el cual tendrá una exactitud de al menos 5 ppm.

El NCR es distribuido con un PID específico dentro del flujo de transporte MPEG-2 que lleva el FLS. La distribución del NCR sigue el mismo mecanismo de distribución del Reloj de Referencia de Programa PCR (*Program Clock Reference*) descrito en el estándar ISO/IEC 13818-1, el cual normalmente se deriva de un codificador de vídeo MPEG, pero para DVB-RCS el NCR se deriva desde el Reloj de Referencia del NCC, el cual tendrá una exactitud de al menos 5 ppm.

2.3.3.1.2 *Sincronización de la portadora*

La portadora MPEG2-TS que transporta el *Forward Link Signalling* contiene la información del NCR, la cual provee una referencia de 27 MHz a los RCSTs, permitiéndoles de esta manera alcanzar una sincronización con respecto a la portadora.

2.3.3.1.3 *Sincronización de la ráfaga*

Otro de los servicios proporcionados por FLS es aquel que le permite a la RCST recuperar la frecuencia central, el tiempo de inicio y la duración de sus ráfagas para la transmisión de información tanto de datos como de señalización.

La forma o Mecanismo de Acceso al Medio en el canal de retorno se resuelve a través de un Acceso Múltiple MF-TDMA (ver sección 2.3.3.7), el cual se especifica por la norma y será visto más adelante con detenimiento en este mismo capítulo y en capítulos posteriores.

Las ráfagas de datos o de señalización asignadas a cada terminal RCST se establecen de acuerdo a un Plan de Asignación de Ráfagas organizado por el NCC, denominado *Terminal Burst Time Plan* (TBTP), y que se recibe, también, a través del FLS.

El TBTP, en palabras simples, le informa a la RCST qué ráfagas o ranuras de tiempo el NCC le ha asignado, y se expresa en términos de la frecuencia

central y el tiempo de inicio absoluto (dado en un valor basado en el contador NCR) de supertramas y de frecuencias asociadas, y tiempos de desplazamientos de asignaciones de ráfagas junto con una descripción de las propiedades de las ranuras de tiempo.

Una supertrama siempre inicia en un valor dado del contador NCR local de la RCST que sirve como una referencia para todas las asignaciones de ráfaga dentro de la supertrama.

Con el propósito de sincronizarse a la red, la RCST reconstruye el valor absoluto del reloj de referencia NCC, para luego comparar el valor reconstruido con el valor NCR dado por el TBTP. La referencia de tiempo para el conteo de ranuras ocurre cuando estos valores son iguales.

El NCR reconstruido de forma ideal deberá ser definido tal y como se observa a la salida de un receptor DVB-S ideal sin retardo. La compensación para el retraso del receptor, si se requiere alcanzar la exactitud especificada, será hecha por la RCST.

2.3.3.1.4 Sincronización del reloj de símbolo

La exactitud del reloj de símbolo estará dentro de las 20 ppm del valor nominal de la velocidad de símbolo, según lo establece la norma. La velocidad del reloj de símbolo tendrá una estabilidad a corto plazo que limitará el error de tiempo de cualquier símbolo dentro de una ráfaga a 1/20 de la duración del símbolo.

2.3.3.2 Formato de ráfaga

El estándar DVB-RCS define cuatro tipos de ráfagas de información presentes en una Trama RCS: ráfagas de tráfico (TRF), de adquisición (ACQ), de sincronización (SYNC) y de señalización de canal común (CSC). Los formatos de estas ráfagas se describen a continuación.

2.3.3.2.1 Formato de la ráfaga de tráfico (TRF)

Las Ráfagas de Tráfico o TRF son usadas para llevar los datos útiles desde la RCST hacia el Gateway(s)/RCST, que para el caso del presente proyecto será el Servidor de Internet. La norma define a su vez dos tipos de ráfagas de tráfico, que son celdas ATM o paquetes MPEG2-TS, siendo este último el que será empleado en el presente proyecto.

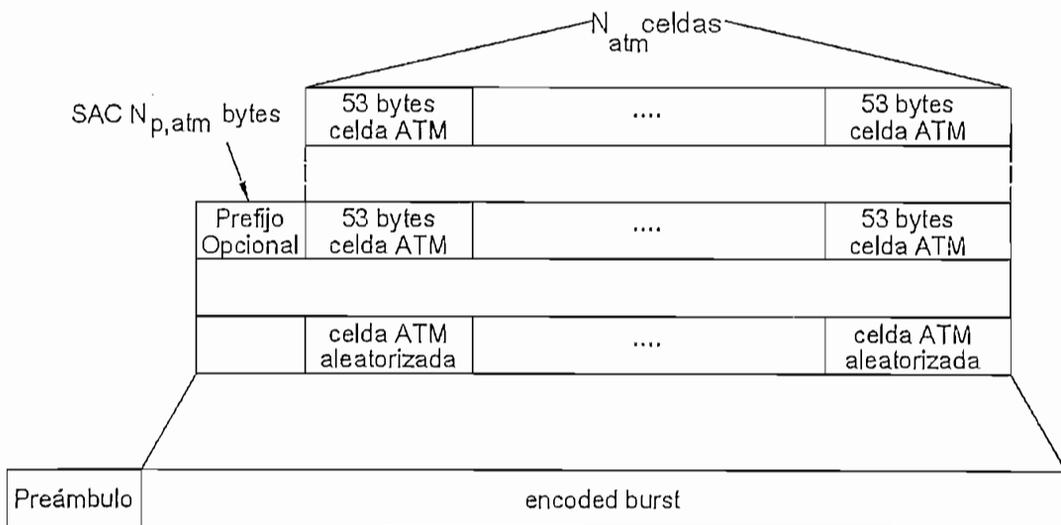


Figura 2.16 Composición de un ráfaga ATM.

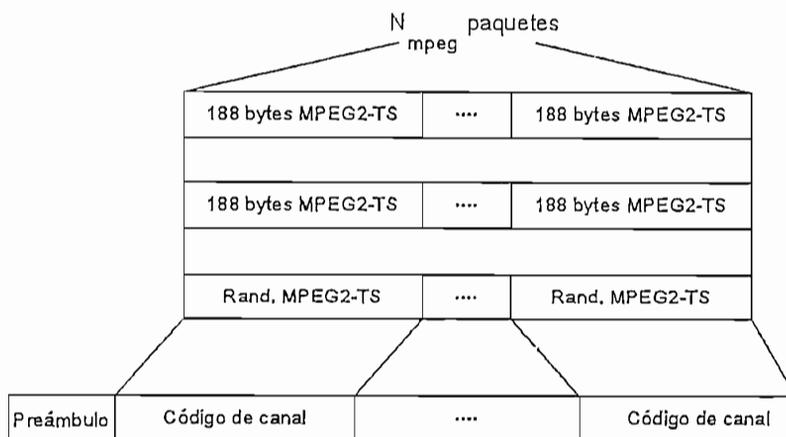


Figura 2.17 Composición de una ráfaga MPEG2-TS.

Cabe señalar también, que un TRF es usualmente seguido por un tiempo de guarda para posibilitar que la RCST disminuya su potencia de transmisión y no interfiera con ráfagas TRF adyacentes.

Las figuras 2.16 y 2.17 muestran la composición de una ráfaga ATM y MPEG2 de tráfico respectivamente.

2.3.3.2.2 *Formato de las Ráfagas de Sincronización (SYNC) y Adquisición (ACQ)*

Las Ráfagas de Sincronización SYNC y de Adquisición ACQ son requeridas para precisar la posición exacta de las ráfagas a ser transmitidas por una RCST durante y después del proceso de entrada (*logon procedure* o TLP).

Es decir, que con el propósito de que una RCST alcance un nivel adecuado de sincronización, el estándar DVB-RCS define dos tipos de ráfagas diferentes (SYNC y ACQ), las cuales son detalladas a continuación.

- Formato de la Ráfaga de Sincronización (SYNC)

Una ráfaga SYNC se utiliza por una RCST con el propósito de mantener la sincronización y también para enviar información de control al sistema.

Las ráfagas SYNC están compuestas por un preámbulo para la detección de la ráfaga, y un byte opcional denominado *SAC_length* el cual es el campo de Control de Acceso al Satélite (SAC).

Después de la aleatorización se puede agregar un código CRC optativo en este campo, resultando un tamaño total de contenedor de *N_p,sync bytes*. Este contenedor es luego protegido con un código de control de error apropiado, tal como CRC o Reed-Solomon.

Al igual que un TRF, un SYNC es seguido por un tiempo de guarda para disminuir la potencia de transmisión de la RCST. La figura 2.18 muestra el formato de la ráfaga SYNC.

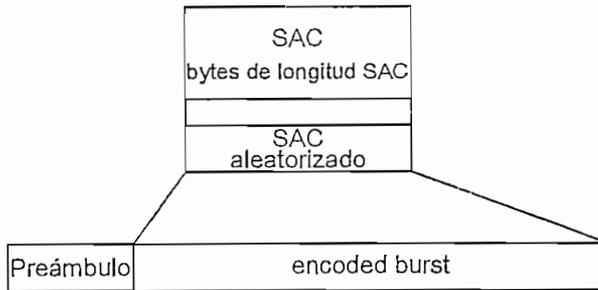


Figura 2.18 Composición de una ráfaga SYNC.

Es importante señalar que la extensión o tamaño de la ráfaga SYNC ha sido dejado abierto por la norma, por lo que dependerá exclusivamente de la capacidad del NCC.

- Ráfaga de adquisición (ACQ)

La ráfaga ACQ le permite a una RCST una sincronización inicial o general, es decir que la ráfaga ACQ puede ser usada para alcanzar la sincronización de la RCST, antes de que éste entre en operación en la red.

El formato de la ráfaga ACQ se muestra en la figura 2.19.

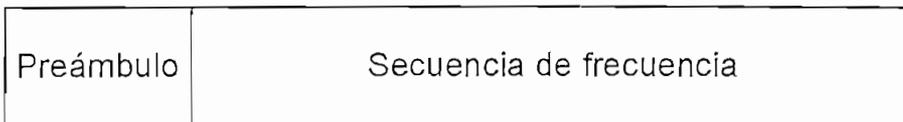


Figura 2.19 Composición de la ráfaga ACQ.

El preámbulo y el contenido (incluida la secuencia de frecuencia), son enviados a las RCSTs a través de la tabla denominada *Timeslot Composition Table*. Cabe señalar que la ráfaga ACQ es, también, seguida por un intervalo o tiempo de guarda.

2.3.3.2.3 *Formato de la Ráfaga de Señalización del Canal Común (CSC)*

Las ráfagas CSC de acceso aleatorio o en contienda, son únicamente usadas por una RCST para identificarse durante el proceso de entrada (ver sección 2.3.5) en la red.

Estas ráfagas están compuestas por un preámbulo para la detección de la ráfaga e inicio de la misma, un campo que describe ciertas características básicas de la RCST útiles para su entrada en la red (tales como sus capacidades, la dirección MAC, soporte de conectividad dinámica, salto de frecuencia), un campo reservado y un identificador del tipo de ráfaga (ver figura 2.20).

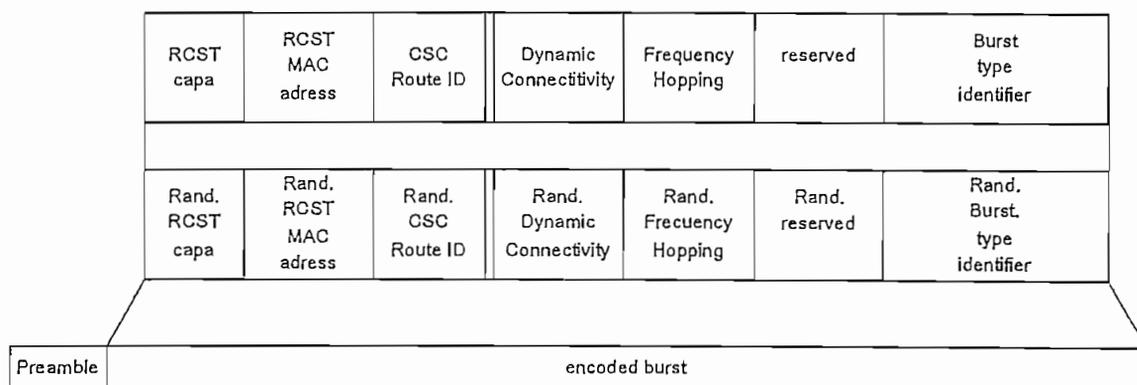


Figura 2.20 Composición de una ráfaga CSC.

La tabla 2.9 muestra el contenido de una ráfaga CSC. Al igual que las anteriores, la ráfaga CSC también es seguida por un intervalo de guarda. La tabla 2.10, por su parte, define los diferentes patrones de bits dentro del campo *RCST Capability* presente en la tabla 2.9. Los 24 bits del campo están numerados desde el LSB (*Less Significant Bit*) hasta el MSB (*More Significant Bit*) usando la notación b0 a b23.

Nombre del campo	Tamaño (bits)	Descripción / Contenido
Preamble	Variable	Preámbulo para detección de ráfaga e inicio de la misma
RCST Capability	24	Ver tabla 2.10
RCST MAC_Address	48	Dirección MAC de la RCST
CSC_Route_ID	16	Permite definir el destino del enlace de ida (<i>Downlink</i>) para la ráfaga CSC en un sistema regenerativo
Dynamic Connectivity	1	"0" cuando la RCST soporta conectividad dinámica y "1" cuando no soporta
Frequency Hopping	1	"1" cuando la RCST soporta salto de frecuencia entre ranuras de tiempo adyacentes, "0" cuando la RCST requiere una ranura TRF entre transmisiones en diferentes frecuencias de portadora
Reserved	21	Reservado
Burst type identifier	1	"1" para la identificación de la ráfaga CSC

Tabla 2.9 Parámetros del campo de datos de la ráfaga CSC.

Parámetro	Tamaño (Bits)	Descripción
Security mechanism	1 (b23)	"1" para implementar mecanismo de seguridad en la RCST. "0" cuando no se implementa
SNMP	1 (b22)	"1" para que la RCST soporte SNMP. "0" en caso contrario
ATM connectivity	1 (b21)	"1" para que la RCST sea capaz de utilizar conectividad ATM (tipo B), "0" cuando ocupa la conectividad del tipo A
MPEG-2 TS TRF	1 (b20)	"1" para que la RCST sea capaz de utilizar MPEG-2 TS TRF. "0" en caso contrario
RCST boards	2 (b19- b18)	Número de receptores RCST para el <i>forward link</i> : "00" para un receptor, "01" para dos, "10" para más de dos, y "11" reservado
RCST ACQ	1 (b17)	"0" para ráfagas ACQ no requeridas, "1" para ACQ requeridas
Multi_IDU	1 (b16)	"0" para configuración de una sola IDU u ODU. "1" cuando dos o más IDUs están conectadas a una sola ODU
SW Versión	8 (b15 – b8)	Dependiente del sistema. Puede ser usado para definir la versión del <i>software</i> de la RCST
Freq Hopping Range	2 (b7 – b6)	Define la ráfaga de la RCST al rango de capacidad de la ráfaga de salto de frecuencia: "00" para 20 MHz, "01" para 120 MHz. Otros patrones dependientes del sistema
MF-TDMA	1 (b5)	"1" para que el RCSt soporte MF-TDMA dinámico. "0" para que la RCST soporte MF-TDMA fijo
RCST Class	2 (b4 – b3)	Dependiente del sistema
Route_ID Capable	1 (b2)	"1" indica que la RCST es capaz de insertar un Route_ID en el campo SAC. "0" caso contrario
RCST Mode	2 (b1 – b0)	"00" para modo de instalación, "01" para modo de operación, "10" para modo de referencia RCST y "11" reservado

Tabla 2.10 Campo *RCST Capability* de la Ráfaga CSC.

2.3.3.3 Aleatorización para Dispersión de Energía

Con el fin de cumplir con las Regulaciones de Radio de la UIT y de asegurar una transición binaria adecuada, el flujo serial de bits de datos en una ráfaga DVB-RCS debe ser aleatorizado. El proceso de aleatorización del flujo de transporte MPEG-2 DVB-RCS es similar al explicado en la sección 2.2.1.4.1.

2.3.3.4 Codificación

Una codificación de canal para protección contra errores se aplica a los datos de control y de tráfico propiamente dichos, lo cuales son transmitidos en los tipos de ráfagas descritos anteriormente. El estándar describe dos esquemas de codificación: *Turbo Code*² y Codificación Concatenada. La RCST podrá implementar ambos esquemas.

Durante una sesión, la RCST no necesita cambiar el esquema de codificación (por ejemplo durante una sesión dada, una RCST usará *Turbo Code* o código concatenado). En el caso de código concatenado, el código de salida es un Código *Reed-Solomon* (no obligatorio) y el de entrada es un Convolutacional No Sistemático (no obligatorio).

Para ambos esquemas de codificación, un CRC (no obligatorio) se puede aplicar en las ráfagas CSC y SYNC con el fin de permitir la detección de errores.

2.3.3.5 Modulación

La señal a ser transmitida, y luego de haber sido codificada, debe ser modulada usando un esquema QPSK, tal como ocurre en DVB-S. El proceso realizado después de la codificación es el indicado en diagrama de bloques de la figura 2.21.

² Una codificación Turbo usa un Código Convolutacional Sistemático Recursivo Circular (CRSC) binario doble.

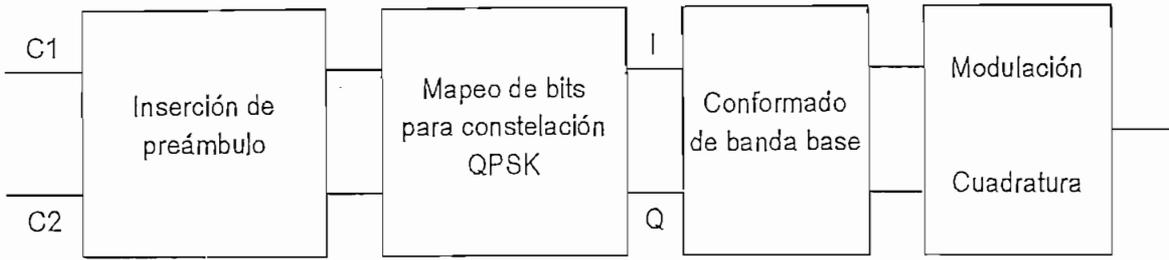


Figura 2.21 Proceso de Modulación

El preámbulo será configurado e indicado al RCST a través de la tabla TCT. Inmediatamente después de la inserción del preámbulo, la salida C1 y C2 del codificador serán enviadas sin modificación al Mapeador de *Bits* o *Bit Mapper* (figura 2.22) QPSK.

Se usa una modulación QPSK convencional en combinación con un código Gray. La asignación de bits en la constelación QPSK será el indicado en la figura 2.22. Si el factor de normalización $1/\sqrt{2}$ se aplica a las componentes I y Q, la energía promedio por símbolo correspondiente, será igual a 1.

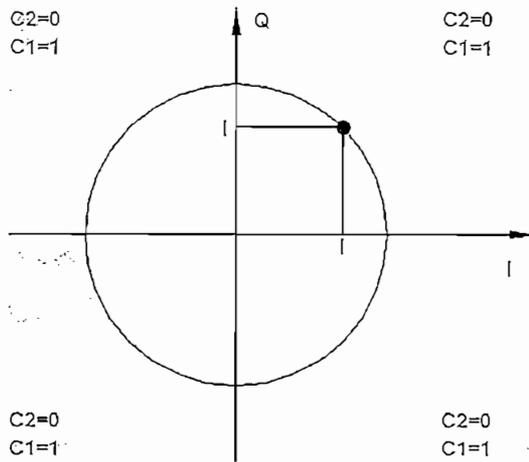


Figura 2.22 Asignación de bits dentro de la constelación QPSK.

La salida C1 del canal de codificación será asignada al canal "I" de la modulación. La salida C2 será asignada al canal "Q" de la modulación. Siguiendo el proceso de modulación indicado, el espectro de la señal QPSK no será invertido.

2.3.3.6.1 Métodos basados en el Campo SAC

Como se mencionó anteriormente, este método transporta las peticiones de capacidad y mensajes M&C a través del campo SAC presente en las ráfagas SYNC y CSYNC, es por eso que se hace necesario el identificar el formato del mismo, así como también su sintaxis y semántica.

La tabla 2.11 muestra los diferentes subcampos presentes en SAC, así como el número de *bits* de cada uno de ellos.

Sintaxis	No. de Bits	
	Reservados	Información
Route_ID		16
Scaling_Factor		1
Capacity_Request_Type		3
Channel_ID		4
Capacity_Request_Value		8
M_and_C_Message		16
Group_ID		8
Logon_ID		16
CNI		8
MODCOD_RQ		8

Tabla 2.11 Sintaxis del campo SAC.

La semántica de los subcampos más relevantes se detalla a continuación.

- Route ID: este campo de 16 *bits* define el enlace *forward* de destino para el prefijo de la carga útil en un sistema regenerativo.
- Capacity Request: cada uno de ellos está compuesto de los siguientes subcampos:
 - o Scaling_Factor: este bit define el factor de escala del subcampo Capacity_Request_value.

Valor	Factor de Escala
0	1
1	16

Tabla 2.12 *Scaling_Factor*.

- o *Capacity_Request_Type*: este subcampo de tres bits especifica el tipo o categoría de la petición de capacidad (ver tabla 2.13). Las categorías de petición de capacidad son descritas en la sección 2.3.3.9.

Valor	Categoría de Capacidad	Significado del <i>Capacity_Request_Value</i>
000	VBDC	Volumen de unidades del tamaño del <i>payload</i> requerido x Factor de escala
001	RBDC	Velocidad de Tx en unidades de 2 kbps requerida x Factor de escala
010	AVBDC	Volumen de unidades del tamaño del <i>payload</i> requerido x Factor de escala
011 - 111	Reservado	

NOTA: El tamaño del *payload* puede ser 53 bytes o 188 bytes dependiendo del modo de encapsulación (ATM o MPEG-2) definido.

Tabla 2.13 *Capacity_Request_Type*.

- o *Channel_ID*: indica el canal para el cual la petición de capacidad es requerida. El valor por defecto es 0000 e indica que la petición es aplicada a cualquier canal.
 - o *Capacity_Request_Value*: este subcampo de 8 bits define el volumen de unidades del tamaño del *payload* o la velocidad de transmisión en múltiplos de 2 kbps de la capacidad requerida.
- *M and C Message*: este subcampo de 16 bits define los mensajes M&C, tal y como se muestra en la tabla 2.14.

M_and_C_Message	Meaning
0x0000	Ningún Mensaje
0x0001	Sincronización Fina alcanzada
0x0002	Petición de <i>Log-off</i>
0x0003 – 0x7FFF	Reservado
0x8000 – 0xFFFF	Repeticiones <i>Echo</i>

Tabla 2.14 M_and_C_Message.

- Group ID: este campo de 8 bits define a qué grupo fue asignada la RCST. El Group_ID y el Logon_ID le permiten al NCC identificar al RCST que ha transmitido una ráfaga en una ranura de tiempo en contención.
- Logon ID: este campo de 16 bits define qué Logon_ID le fue asignado a la RCST en el proceso de *logon* (ver sección 2.3.5.3).

Una vez identificado el formato del campo SAC y las funciones de cada uno de sus subcampos, es tiempo de revisar lo que dice el estándar acerca de los cuatro métodos de mensajes MAC basados en el campo SAC.

- Método del Prefijo: Este mecanismo está basado en los bytes opcionales de prefijo asociados a las ráfagas ATM. Si se usa, el prefijo lleva la información de control y manejo desde la RCST al NCC. Este mecanismo es respaldado por los subcampos SAC: route_ID y de *request*.
- Método de Mini Ranuras: Este mecanismo se basa en una asignación periódica a las RCST de ráfagas más pequeñas (mini ranuras SYNC) que las ranuras de tiempo TRF. Éste lleva la información de control y manejo desde la RCST al NCC y se usa también para mantener la sincronización de la RCST. Este mecanismo es respaldado por los subcampos *request* SAC que se usan en las ráfagas SYNC.
- Método de Contención basado en Mini Ranuras: Este mecanismo es el mismo que el método anterior, pero con la diferencia de que las mini ranuras pueden

ser accesadas por cualquier RCST en un modo de contienda. Este mecanismo es respaldado por los subcampos SAC: *request*, *Group_ID* y *Logon_ID* usados en las ráfagas SYNC en contienda o CSYNC.

- Método de Adaptación del Campo MPEG (MPAF): Este método es opcional dentro de la norma y se basa en el uso de los *bytes* de datos privados de la capa de adaptación de MPEG para transportar los mensajes MAC. El problema con este método es que la disponibilidad de éste no es señalizada por las tablas propias de DVB si no más bien por la misma cabecera MPEG, lo que podría provocar una incompatibilidad en aquellos otros dispositivos de la red que no estén implementado esta opción, como por ejemplo los *gateways*.

2.3.3.6.2 Método DULM para la entrega de mensajes MAC

EL Método de Etiquetado de Unidades de Datos o *Data Unit Labelling Method (DULM)* es un método basado en mensajes que permite a las RCSTs transmitir información de control y manejo al NCC en el *payload* de las unidades de datos ya asignadas a ellos en las ráfagas TRF.

Esas unidades de datos pueden ser celdas ATM o paquetes MPEG-2, dependiendo del tipo de encapsulamiento escogido en un principio para la RCST. El método DULM se puede combinar con las otras técnicas de transporte de mensajes MAC, todo depende de cómo esté configurada la red.

El uso de DULM puede responder a una iniciativa del NCC o de la RCST. De hecho, una RCST activa puede insertar peticiones de ancho de banda en las ranuras de tráfico ya asignadas a ella.

Como ya fue indicado, el método DULM puede ser utilizado tanto para encapsulado ATM como para encapsulado MPEG, por supuesto la estructura no será la misma para ambos. Es por eso que a continuación se hace una distinción entre el método DULM aplicado a ATM y el método DULM aplicado MPEG-2.

- Método DULM con el formato ATM: Para las RCSTs que usan ráfagas TRF ATM, la utilización del método DULM será identificado con un único valor en la cabecera de la Unidad de Datos (celda para ATM), con el MSB del campo PT puesto en '1', de esta forma se permite la discriminación de los mensajes DULM del resto de la información del tráfico normal.

Un mensaje DULM estará compuesto por un número entero (entre 1 y 64) de Elementos de Información o IE (*Information Element*). Cada IE deberá cumplir con el formato descrito en la figura 2.23, y estar constituidos por dos *bytes* de cabecera (Tipo de IE y Longitud de IE) más un cuerpo de *n bytes*, con *n* entre 1 y 512. Un mensaje DULM dado puede estar compuesto por IEs de diferentes tamaños.

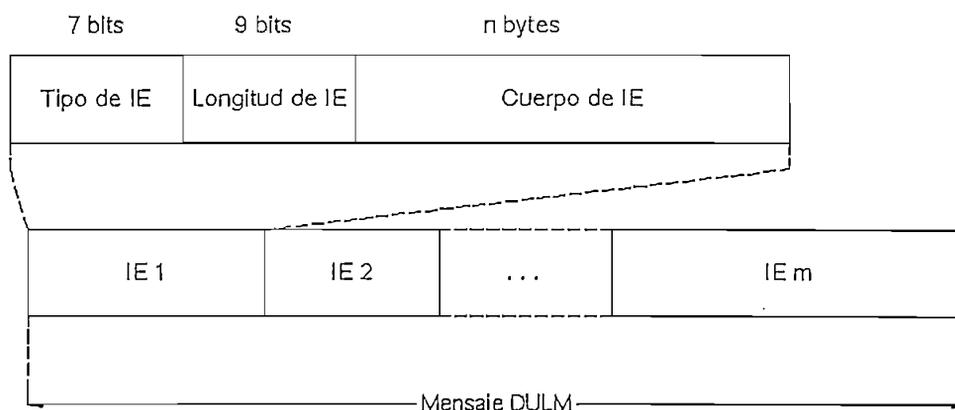


Figura 2.23 Formato del Mensaje DULM con el formato ATM.

- Método DULM con el formato MPEG

Para los RCTSs que usan las ráfagas TRF MPEG, los mensajes DULM embebidos en ellas serán distinguidos por un único PID (también conocido como CTRL/MGNM PID) característico en la cabecera de dichas ráfagas. Este PID es obtenido por la RCST durante el procedimiento de *logon* o de entrada a la red.

Los mensajes DULM utilizarán el mecanismo de transporte descrito a continuación y que es ilustrado en la figura 2.24.

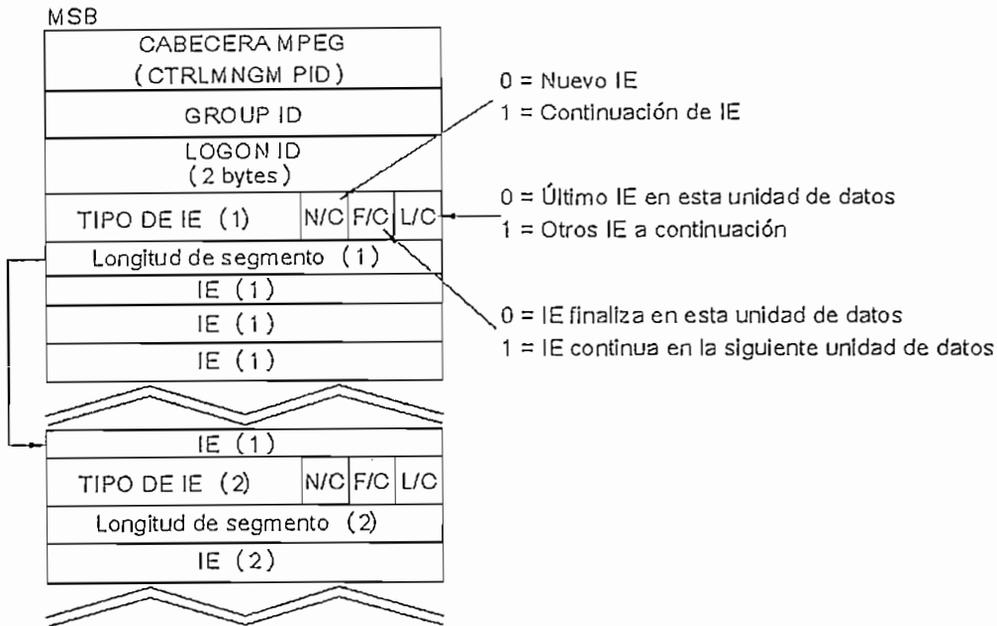


Figura 2.24 DULM sobre paquetes MPEG.

Los primeros 3 bytes de cada carga útil de las Unidades de Datos corresponderán a los Group_ID y Logon_ID de la RCST, entonces uno o más IEs serán insertados en el paquete MPEG siguiendo la estructura descrita a continuación.

El primer byte de un IE contiene los cuatro campos siguientes:

- El campo [IETYPE] está contenido en los primeros 5 bits. Éste indica el tipo de IE presente de acuerdo a la tabla 2.15.
- El campo [N/C] está contenido en el sexto bit. Si [N/C]=0 el IE está comenzando en esa Unidad de Datos, si [N/C]=1 el IE es la continuación de una Unidad de Datos CTRL/MNGM previa de la misma RCTS.

- El campo [F/C] está contenido en el séptimo *bit*. Si [F/C]=0 el IE terminará en esa Unidad de Datos, si [F/C]=1 el IE continuará en la siguiente Unidad de Datos CTRL/MNGM de la misma RCTS.
- El campo [L/C] está contenido en el octavo *bit*. Si [L/C]=0 el IE es el último de la Unidad de Datos CTRL/MNGM, si [L/C]=1 otro IE sigue en la misma Unidad de Datos CTRL/MNGM.

El segundo *byte*, por su parte, contiene el campo [SLENG]. Éste indica la longitud del segmento en número de *bytes*, definido como la parte del IE contenida en la Unidad de Datos actual. Los siguientes *bytes* contienen los fragmentos de longitud variable del IE.

Tipo de IE MPEG		Longitud del IE
0x00	Petición de Capacidad	2 bytes
0x01	M&C	2 bytes
0x02	Reservado	
0x03	Mensaje <i>Header</i>	4 bytes
0x04	Causa	2 bytes
0x05	<i>Channel_ID</i>	1 bytes
0x06	Dirección Origen	6 bytes
0x07	Dirección Destino	6 bytes
0x08	Identificador del flujo de ida	3 bytes
0x09	Identificador del flujo de retorno	3 bytes
0x0A	Tipo	1 bytes
0x0B	Perfil de ida	3 bytes
0x0C	Perfil de regreso	3 bytes
0x0D	Respuesta de seguridad	8 bytes
0x0E – 0x10	Reservado	
0x11	Respuesta de cambio de llave principal	
0x12	Reservado	
0x13	Respuesta de cambio de llave rápida	
0x14	Reservado	
0x15	Respuesta de cambio de llave explícita	
0x16	ACM	2 bytes
0x17 – 0x1E	Reservado	
0x1F	Espera	

Tabla 2.15 Tipos de IE en MPEG.

2.3.3.7 Acceso múltiple MF-TDMA

En una red interactiva DVB-S/RCS el acceso al medio, por parte de las RCSTs, se lo logra a través de un esquema de Acceso Múltiple por División de Tiempo Multifrecuencial o MF-TDMA (*Multiple Frequency Time Division Multiple Access*).

Este esquema de acceso permite a un grupo de RCSTs comunicarse con un *Gateway* (que puede ser un ISP o el mismo NCC) utilizando un grupo de frecuencias portadoras, cada una de las cuales está dividida en ranuras de tiempo.

El NCC asignará a cada RCST activa una serie de ráfagas, cada una definida por una frecuencia, un ancho de banda, un tiempo de inicio y uno de duración.

Se debe indicar también que la norma contempla dos variantes del método MF-TDMA: un MF-TDMA fijo, y un MFTDMA dinámico. El criterio de selección de uno u otro esquema dependerá de la versatilidad que se desea dar a la red en cuanto al manejo de recursos. Como sea, a continuación se describen ambos métodos.

2.3.3.7.1 MF-TDMA Fijo

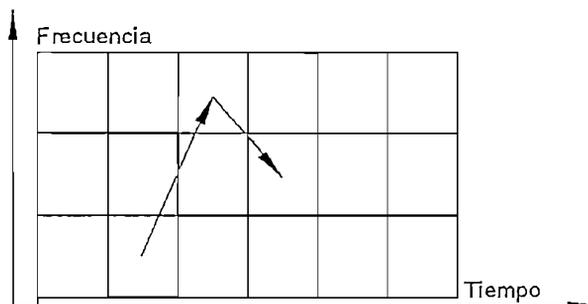


Figura 2.25 MF-TDMA de ranuras fijas.

En el esquema MF-TDMA de ranuras fijas, el ancho de banda y la duración de las ranuras de tráfico sucesivas usadas por la RCST son fijos, tal como se puede observar en la figura 2.25 donde la flecha indica una secuencia típica de ranuras asignadas por el NCC a una RCST.

2.3.3.7.2 MF-TDMA Dinámico

El esquema MF-TDMA de ranuras dinámicas, utiliza una flexibilidad adicional en las RCSTs para variar el ancho de banda y la duración de ranuras sucesivas asignadas a una RCST. Además de cambiar la frecuencia de portadora y la duración de la ráfaga, la RCST puede también cambiar la velocidad de transmisión y de codificación entre ráfagas sucesivas.

La ventaja de esta gran flexibilidad en una RCST se traduce en una mayor eficiencia en la adaptación a los extensamente cambiantes requerimientos de transmisión típicos de las aplicaciones multimedia.

El principio básico de dicha flexibilidad de una RCST se ilustra en la figura 2.26, donde la flecha muestra una RCST usando ranuras sucesivas con diferentes anchos de banda y duración.

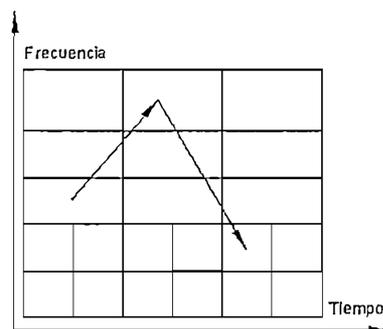


Figura 2.26 MF-TDMA de ranuras dinámicas, usando una RCST flexible.

Las RCSTs tienen un rango específico de frecuencia, para el salto de frecuencia, de ranura a ranura. Este rango de salto de frecuencia es comunicado

desde la RCST al NCC dentro de la ráfaga CSC (ver sección 2.3.3.2.3) durante el procedimiento de entrada o *logon* (ver sección 2.3.5.3).

La capacidad del salto de frecuencia para una RCST individual queda a criterio del fabricante pero deberá ser de al menos 20 MHz (por ejemplo ± 10 MHz alrededor de la frecuencia central) para evitar interferencias.

2.3.3.8 Segmentación de la capacidad del enlace de retorno

En una Red Satelital Interactiva, las ranuras de tiempo del enlace de retorno están organizadas y numeradas de tal manera que la red pueda asignarlas a RCSTs individuales. Con este propósito, los recursos de la Red Interactiva están organizados en Supertramas, Tramas y Ranuras de Tiempo (*Time Slots*).

2.3.3.8.1 Supertramas

Una supertrama es una porción de tiempo y frecuencia en el enlace de retorno y está siempre asociada a un identificador *Superframe_ID*. Dentro de las Redes Satelitales Interactivas, un *Superframe_ID* identifica los recursos del enlace de retorno accedidos por un grupo de RCSTs. La figura 2.27 muestra un típico ejemplo por el cual los *Superframe_IDs* son realmente grupos separados de frecuencias portadoras.

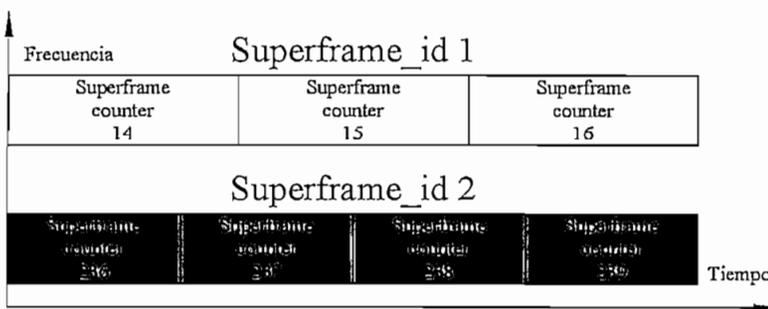


Figura 2.27 Ejemplo típico de supertramas de una Red Satelital Interactiva.

En un Red Satelital Interactiva, la capacidad global del enlace de retorno puede ser segmentada en medio de grupos de RCSTs, y la red administrará entonces, separadamente algunos *Superframe_IDs*.

Como se muestra en la figura 2.27, las supertramas consecutivas de un *Superframe_ID* dado, son contiguas en el tiempo. Cada suceso de una supertrama en el tiempo es etiquetado con un número denominado *Superframe_counter*.

Para cada supertrama (de un *Superframe_ID* dado), la asignación de las ranuras de tiempo es comunicada a las RCSTs a través de la tabla *Terminal Burst Time Plan (TBTP)* (ver sección 2.3.8.6). Una RCST es permitida transmitir ráfagas solo en las ranuras de tiempo que fueron asignadas a ella (acceso dedicado), o en ranuras de tiempo accesadas aleatoriamente (acceso en contención CSC y CSYNC).

La duración de la supertrama es por lo tanto el período elemental de tiempo que se utiliza para la asignación de recursos a los terminales.

2.3.3.8.2 *Tramas*

Una supertrama está compuesta de tramas, y a su vez, éstas están compuestas por ranuras de tiempo. La trama está en un nivel intermedio entre las supertramas y las ranuras de tiempo.

Las tramas son introducidas por motivos de eficiencia de señalización (en el enlace de señalización de ida). La duración de una trama no se utiliza como la base de cualquier proceso de asignación de ranuras de tiempo.

En una supertrama, las tramas son numeradas desde 0 (la frecuencia más baja, primera en el tiempo) hasta N (la frecuencia más alta, última en el tiempo), ordenadas en tiempo y después en frecuencia como se muestra en la figura 2.28. N será menor o igual que 31.

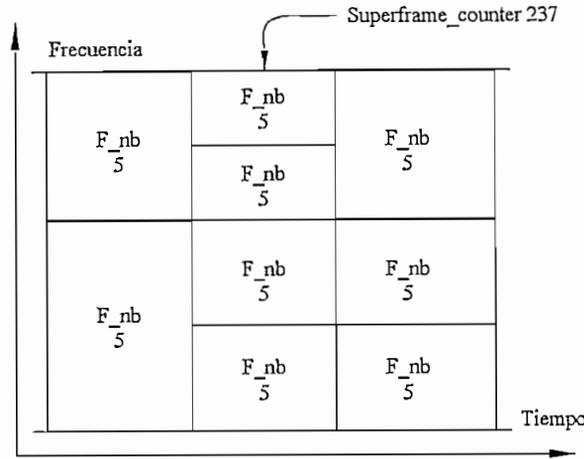


Figura 2.28. Ejemplo de composición de una supertrama.

Las tramas de una supertrama no necesariamente deben todas tener la misma duración, ancho de banda y composición de ranuras de tiempo. Las tramas y las supertramas pueden todas tener la misma duración, en el caso de que las tramas sean vistas como sub-bandas de frecuencia de la supertrama.

De cualquier manera esta propiedad no es mandatoria y la figura 2.28 muestra un ejemplo de una supertrama durando 3 veces más que cada una de sus tramas.

2.3.3.8.3 Ranuras de tiempo

Una trama está compuesta por ranuras de tiempo. Un *frame_id* (valor que identifica una trama) identifica un arreglo particular de ranuras de tiempo. Por ejemplo, un *frame_id* = 1 podría identificar una secuencia de 10 ranuras de tiempo de tráfico de usuario en la misma portadora, y el *frame_id* = 2 una secuencia de 4 ranuras de tiempo de control seguidas por 8 ranuras de tiempo de tráfico de usuario, todas en la misma portadora.

Una trama puede abarcar algunas frecuencias portadoras. En una trama, las ranuras de tiempo son numeradas desde 0 (frecuencia más baja, primera en el tiempo) hasta M (frecuencia más alta, última en el tiempo), ordenadas en

tiempo y después en frecuencia como se muestra en la figura 2.29. M deberá ser menor o igual a 2 047.

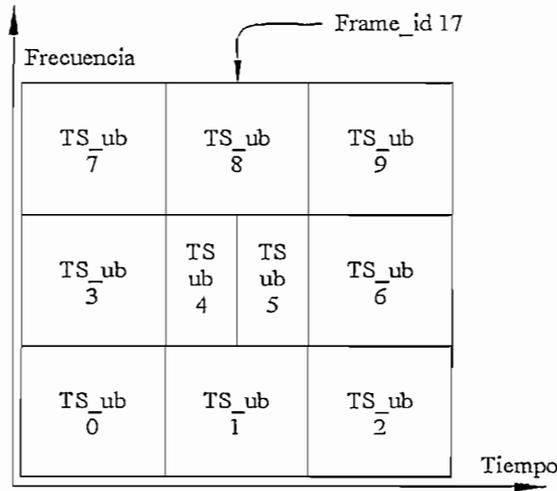


Figura 2.29 Ejemplo de composición de una trama.

Para propósitos de asignación de recursos, cada ranura de tiempo se identifica, únicamente, por su *Superframe_ID*, *Superframe_counter*, *Frame_number* y *Timeslot_number* todos ellos presentes en el TBTP.

La RCST procesará entonces, el mensaje TBTP desde el NCC para su área de distribución, con el fin de extraer la cuenta de asignación y la distribución de las ranuras de tiempo para sus próximas transmisiones en el *uplink RCS*.

El tiempo de latencia desde la llegada del mensaje TBTP al RCST hasta que la RCST esté lista a transmitir las ráfagas asignadas por el mensaje TBTP no excederá los 90 ms.

2.3.3.9 Categorías de Peticiones de Capacidad

Como ya se mencionó anteriormente, una RCST, antes de poder transmitir sus datos, debe realizar una petición de asignación de recursos al NCC. En la sección 2.3.3.6 se revisó los distintos métodos o mecanismo para la entrega de los mensajes MAC de petición de capacidad. Ahora resta revisar las “Categorías o

Tipos de Peticiones de Capacidad” contempladas por la norma DVB-RCS que son llevados en los mensajes MAC.

Es así que el estándar especifica los siguientes tipos de asignación de capacidad: CRA (*Constant Rate Assignment*), RBDC (*Rate Based Dynamic Capacity*), VBDC (*Volume-Based Dynamic Capacity*), AVBDC (*Absolute Volume Based Dynamic Assignment*) y FCA (*Free Capacity Assignment*). [7]

2.3.3.9.1 *Asignación de tasa continua CRA*

En la asignación de tasa continua CRA el terminal manifiesta su necesidad de capacidad RCS al momento del establecimiento de la conexión, en términos de un número fijo de ranuras por trama.

Si la conexión es admitida, el terminal tendrá reservadas el número de ranuras solicitadas por cada trama durante la conexión. De esta forma el tráfico no está sujeto a ningún tipo de retraso por planificación o encolamiento.

El NCC deberá asegurar que todas las asignaciones del tipo CRA no excedan la capacidad total del sistema.

Este tipo de asignación está orientada hacia conexiones de tiempo real como las CBR, que necesitan de una asignación fija garantizada con retardo y jitter mínimos. Esta categoría es también útil para tráfico de tiempo real con tasa variable (VBR-RT) que no puede tolerar el retardo mínimo por planificación MSL (*Minimum Scheduling Latency Delay*).

2.3.3.9.2 *Asignación de capacidad basada en tasa RBDC*

Para la asignación del tipo RBDC, la estación realiza una petición de un número variable de ranuras por trama, hasta un máximo valor negociado al momento del establecimiento de la conexión. La petición se mantiene efectiva durante un número determinado de tramas siempre y cuando la estación actualice

la petición o hasta alcanzar un tiempo límite de asignación, de esta forma se evita que una determinada estación monopolice la utilización de los recursos.

La diferencia con la asignación CRA, es que la estrategia RBDC permite realizar una multiplexación estadística entre los terminales, es decir, la asignación de recursos es dinámica lo que resulta en un uso más eficiente de la capacidad del sistema.

Por otra parte y al igual que en el caso de la asignación CRA, el NCC debe asegurar que la capacidad total CRA y RBDC de todos los terminales no exceda la capacidad total del sistema.

Esta asignación puede ser usada en conexiones con tolerancia al retardo, es decir, en conexiones que no son de tiempo real como las que soportan servicios de VoIP, por ejemplo. La asignación CRA y RBDC pueden usarse en combinación, con la CRA como una capacidad mínima fija por supertrama, y la asignación RBDC como componente de variación dinámica sobre la CRA.

2.3.3.9.3 Asignación de capacidad basada en volumen VBDC y AVBDC

En la asignación VBDC, las estaciones envían sus peticiones de capacidad en términos del número de recursos RCS necesarios para vaciar sus colas de transmisión. Tales requerimientos pueden complementarse con posteriores peticiones, en cuanto exista tráfico encolado que no tenga asociado una petición de capacidad.

El programador de asignación del NCC acumula todas las peticiones de la estación pero no ofrece ningún tipo de garantía sobre la disponibilidad de la capacidad.

La capacidad asignada para las peticiones VBDC, se encontrará dentro de las limitaciones resultantes después de las asignaciones CRA y RBDC que mantenga la estación y que están sujetas a su capacidad máxima de transmisión.

Las peticiones de capacidad VBDC están enfocadas hacia conexiones tolerantes a la variación del retardo y con tráfico a ráfagas y permite manejar mejor la multiplexación estadística y por tanto un uso más eficiente de los recursos.

La asignación VBDC es aplicable a servicios que no son tolerantes al retardo como por ejemplo el tráfico de navegación Web.

De la misma forma que las peticiones VBDC, los requerimientos del tipo AVBDC permiten también a las estaciones el realizar una petición dinámica de recursos en función de las colas de tráfico y se usa en lugar de la VBDC cuando la RCST detecta que una petición VBDC se ha perdido, como por ejemplo en el caso de usar CSYNC como método para llevar las peticiones de capacidad.

En otras palabras, la categoría de capacidad AVBDC representa el valor total del volumen de tráfico que tiene una estación, mientras que las VBDC corresponden a los incrementos de volumen de tráfico desde la última petición de este tipo, realizada por la estación.

2.3.3.9.4 *Asignación de capacidad disponible FCA*

La asignación FCA permite al programador de asignación de recursos del NCC maximizar la utilización de los mismos distribuyendo la capacidad no asignada entre un subconjunto predeterminado de RCSTs.

Esta asignación de capacidad deberá ser automática por lo que no implica ningún tipo de señalización desde las RCST al NCC. La asignación de capacidad de esta categoría será considerada como un "extra" que puede reducir los retrasos de aplicaciones tolerantes al *jitter*.

El NCC por tanto deberá en primer lugar satisfacer todas las demás categorías de petición de recursos antes de realizar una asignación FCA.

2.3.4 MÚLTIPLES CANALES POR TERMINAL

Dentro de una misma Multitrama MF-TDMA, se establece una diferenciación en la asignación de recursos mediante el uso de un conjunto de identificadores de canal (un canal por cada servicio). De esta forma, una estación RCST es capaz de distinguir la asignación de recursos recibida, tal y como si se tratara de una asignación a terminales distintos.

Los canales pueden también manejarse de forma separada de acuerdo con el perfil de tráfico o de la red, es decir, la estación podría mantener una asociación directa entre el identificador de canal y el tipo de flujo a transmitirse.

La asignación de recursos físicos que recibe una RCST corresponde a un conjunto de ranuras MF-TDMA de la supertrama(s) a la que tiene acceso. Como ya se ha mencionado, esta asignación de recursos puede agruparse en distintos canales que constituyen caminos independientes por lo que las asignaciones y peticiones de capacidad se realizan de forma independiente.

La figura 2.30 muestra una representación funcional del manejo de los canales dentro de la norma DVB-RCS.

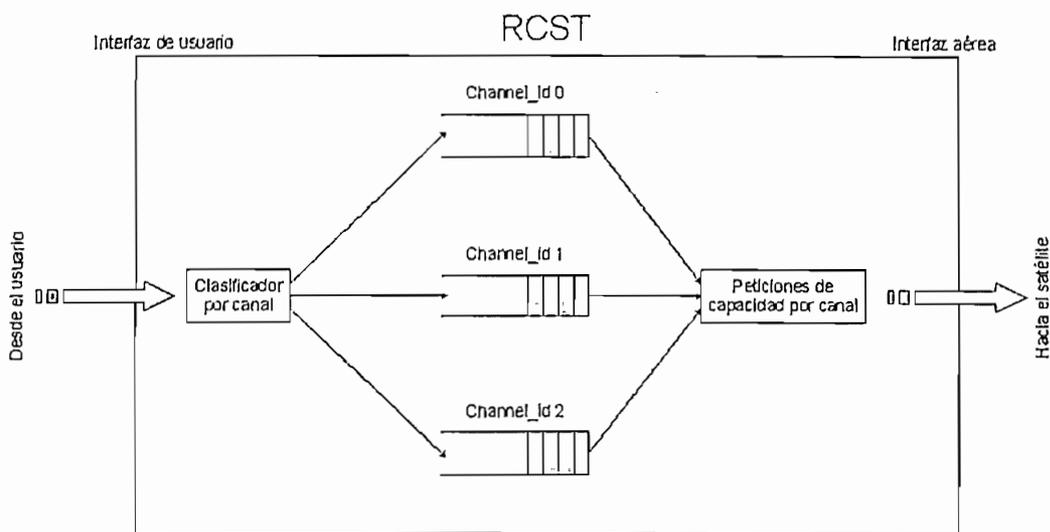


Figura 2.30 Representación funcional del uso de los canales.

La norma RCS identifica a los canales a través de un *channel_id*. La asignación de recursos hecha por el NCC específicamente a un canal se indica a través de la tabla TBTP, mediante el campo *Multiple_channels_flag*. Caso contrario, la RCST asocia un *channel_id* = 0 para la asignación. La asignación de identificadores de canal diferentes está bajo control del administrador de la red pudiendo existir hasta 16 *channel_id* distintos.

Las principales razones que motivan el manejo de la asignación de recursos RCS en distintos canales por terminal, son las siguientes:

- Provisión de diferentes clases de calidad de servicio QoS. Por ejemplo, si se tiene un canal asignado para tráfico "*best effort*" y un canal para "tráfico de primera clase" en cada RCST, la red podría asignar las ranuras de tiempo RCS dando prioridad a un tipo de canal sobre el otro.
- Provisión de servicios orientados a conexión (como circuitos virtuales ATM) donde puede ofrecerse garantías de servicio por circuito virtual (CV). Los CV estarían asociados entonces a los identificadores de canal.
- Provisión de comunicaciones simultáneas entre una RCST y varias pasarelas. Esta característica es precisamente el caso de las arquitecturas multiestrella, donde las RCST pueden acceder a más de un proveedor de servicio cuyos recursos del canal de retorno tienen asociado un identificador independiente.

Si se define que una RCST establezca una conexión con otra antes de enviar datos, esta conexión puede tener asociado un identificador de canal que puede ser único para un conjunto de RCSTs.

De esta forma un canal podría cursar una o varias conexiones entre diferentes RCSTs destino de forma agregada, siendo abierto implícitamente en el momento de la petición de transporte de datos hecha por la RCST fuente, y cerrarse cuando la última conexión que utilizaba el canal se cierra.

La posibilidad de agregación deja el control de la repartición de recursos sobre un mismo canal bajo responsabilidad de la RCST, que podría realizar una multiplexación estadística a fin de aprovechar mejor los recursos no utilizados y que fueron solicitados por una conexión con fuertes restricciones de ancho de banda y retardo, pero con tráfico a ráfagas o de tasa variable (VBR).

2.3.5 PROCEDIMIENTOS DE SINCRONIZACIÓN

En esta sección se definirán los Procedimientos de Entrada conocidos también como *Terminal Logon Procedure* (TLP), que permiten al RCST acceder a la Red Satelital Interactiva así como también, cómo puede ésta abandonarla (procedimiento de salida *logoff*).

Se debe mencionar que, según lo establece la norma, el período de tiempo desde que el terminal se registra en la red hasta que la abandona es denominado sesión.

2.3.5.1 Secuencia global de eventos

Con el fin de permitir el procedimiento de entrada, la RCST deberá encontrarse en un estado llamado *Receive Synchronization State*, el cual se alcanza siguiendo el "Procedimiento Inicial de Sincronización" que será descrito en la sección 2.3.5.2.

La entrada de una RCST dentro del sistema es entonces alcanzada siguiendo las cuatro fases indicadas a continuación y que más adelante serán tratadas a profundidad:

- Procedimiento de entrada o TLP: La RCST solicita un acceso inicial a la red obteniendo información inicial de entrada de la misma (o alternativamente el pedido de entrada puede ser rechazado por la red).

- Procedimiento de adquisición de sincronización gruesa (*Acquisition coarse synchronization procedure*): Es un procedimiento opcional que le permite a la RCST mejorar su sincronización a nivel físico (ajustes de frecuencia, tiempo y potencia).
- Procedimiento de sincronización fina (*Fine Synchronization Procedure*): Es un procedimiento también opcional que permite a la RCST completar su sincronización física.
- Procedimiento para mantener la sincronización: Describe el proceso para que la RCST mantenga su sincronización física durante toda la sesión.

Dependiendo de los procedimientos, la RCST puede estar en uno de los siguientes estados:

- *Hold State*: la RCST se encuentra en el modo *hold*, después de haber sido apagado o reiniciado. Un terminal va desde el estado *hold* al estado de Recepción de Sincronización (*Receive Sync*) solo cuando éste recibe una tabla TIM (*Terminal Information Message*), con la bandera *Transmit_Disable* en el valor "0" (ver sección 2.3.8.7).
- *Inactive Off/Stand-by State*: la RCST no está prendida, se encuentra en un estado de espera o ha perdido sincronismo.
- *Receive sync state*: La RCST ha adquirido el enlace de ida (*forward link*).
- *Ready for coarse sync state*: La RCST ha sido detectada por el NCC, y puede iniciar el procedimiento de sincronización gruesa.
- *Ready for fine sync state*: La RCST ha sido detectada por el NCC, y puede iniciar el proceso de sincronización fina.
- *Fine sync state*: La RCST está sincronizada y puede enviar tráfico.

El procedimiento de salida o *logoff*, por su parte, es descrito en la sección 2.3.5.7, y básicamente provee el mecanismo para abandonar la red a la RCST.

Las figuras 2.31 y 2.32 dan una visión global de la secuencia que una RCST sigue para estar habilitada y poder enviar tráfico dentro de la red. En la figura 2.31 se muestra, a manera de un diagrama de flujo, el proceso de entrada y sincronización de la RCST, donde el cuadro  representa los posibles estados de la RCST y  representa los procedimientos descritos anteriormente.

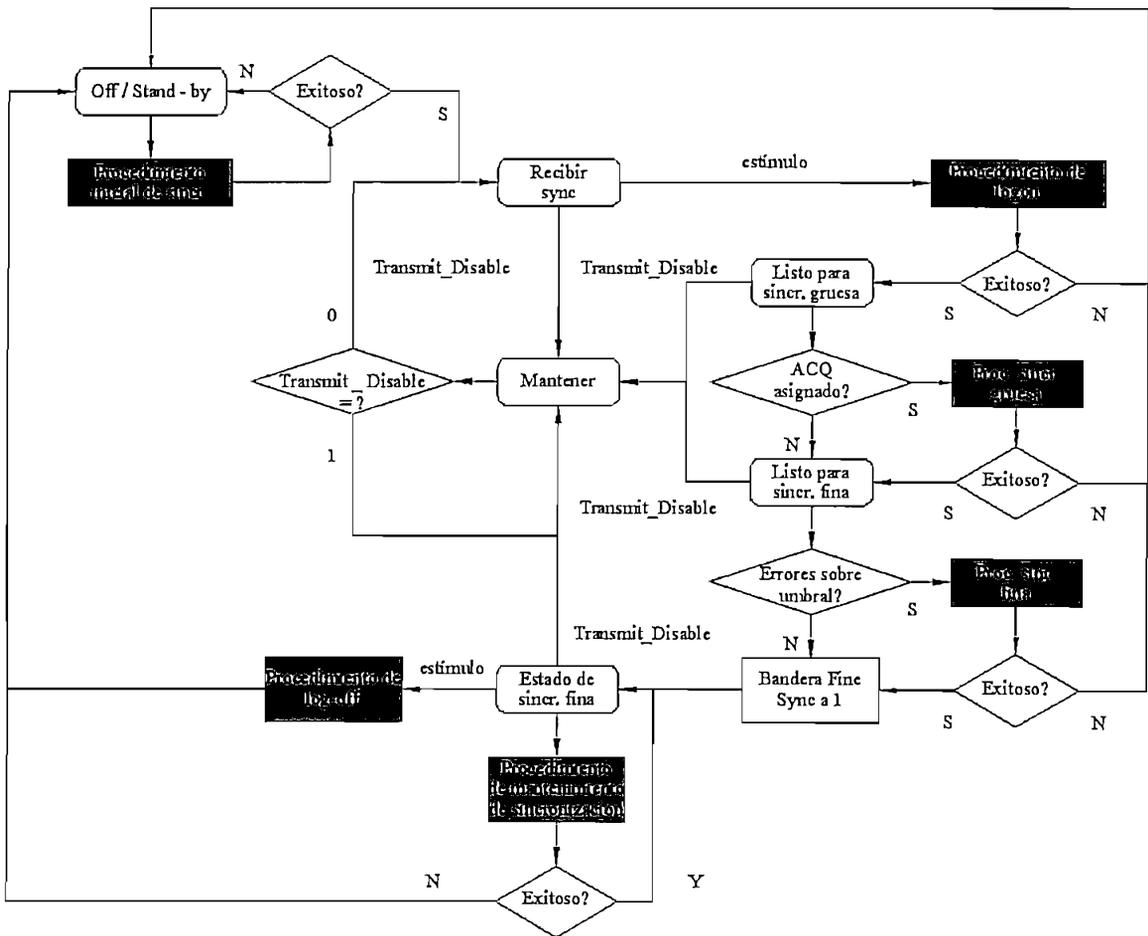


Figura 2.31 Diagrama del estado del procedimiento de entrada y sincronización de una RCST.

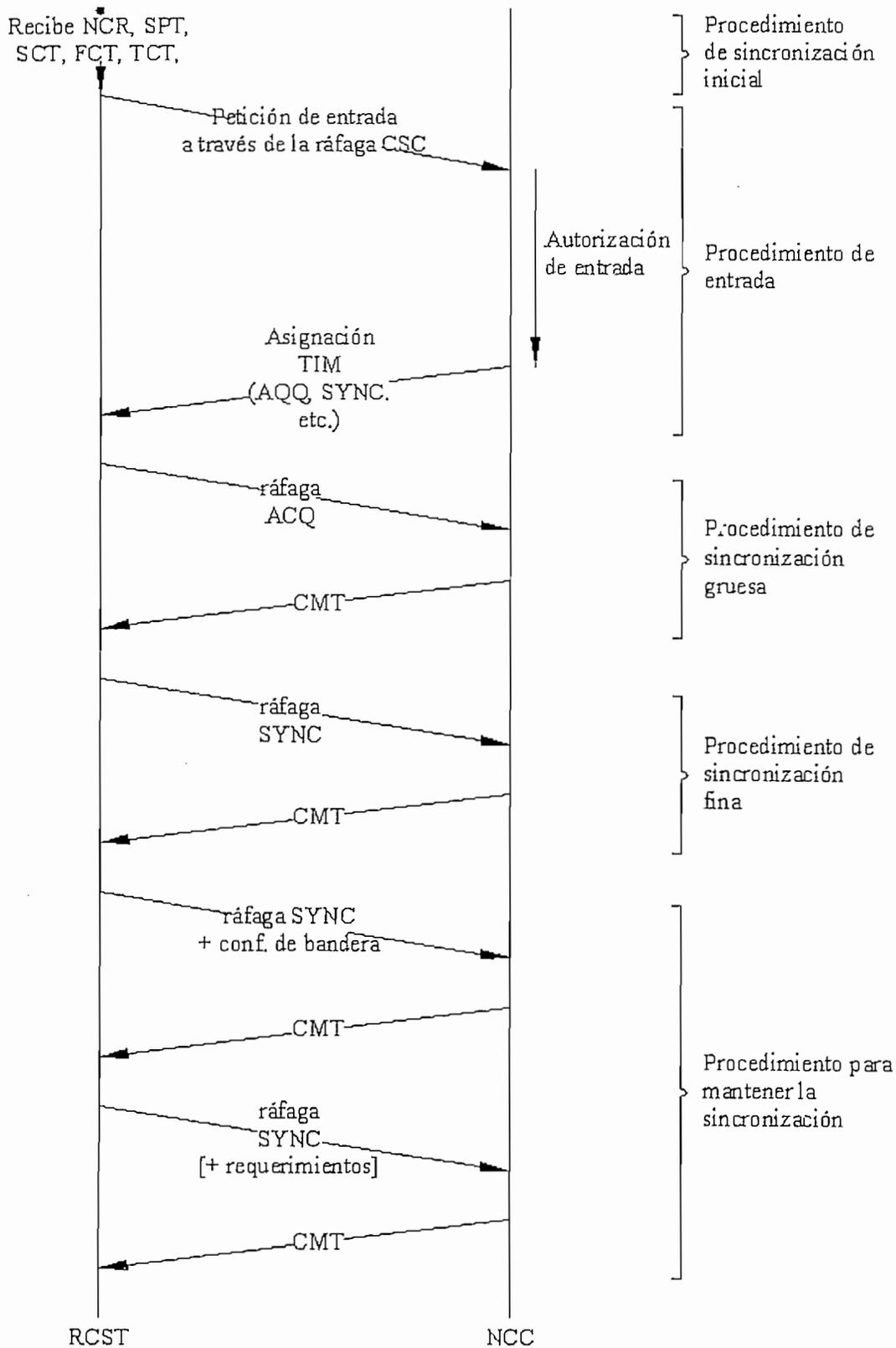


Figura 2.32 Ejemplo del flujo de señalización de entrada entre el NCC y la RCST.

Cabe señalar, que una RCST no está en condiciones de transmitir ráfagas TRF hasta que haya alcanzado el estado de sincronización fina (*Fine Sync*).

Por otra parte, el intercambio de mensajes de señalización entre la RCST y el NCC durante el proceso de *Logon*, incluido mensajes opcionales, se ilustra en la figura 2.32. Esta figura muestra el flujo normal de eventos en el caso de que sean usados los procedimientos de sincronización fina y gruesa.

2.3.5.2 Procedimiento Inicial de Sincronización

Después de haber sido encendido, la RCST deberá proceder como se detalla a continuación:

- La RCST primero debe seguir el procedimiento descrito en la sección 2.3.9 para buscar toda la información de control necesaria relacionada con la operación de la red RCS. Esta información incluye sincronización NCR, a través de la cual la RCST inicializa su reloj interno sincronizándolo al reloj NCR el cual se transmite por el NCC en el enlace de ida.
- Después de alcanzado el procedimiento anterior, la RCST calculará los rangos satelitales para ambos enlaces, para el de ida y el de retorno, usando los datos satelitales contenidos dentro de la *Satellite Position Table* (SPT, ver sección 2.3.8.4) más el conocimiento de su propia ubicación (latitud, longitud y altura sobre el nivel del mar). La RCST usará estos rangos para calcular los retardos de propagación correspondientes del satélite a la RCST y viceversa. La posición nominal del satélite, la cual se puede encontrar en la NIT, se utilizará en el caso de que el NCC no transmita la SPT.
- La RCST continuará recibiendo el NCR a lo largo de toda la sesión. En el caso de que la sincronización NCR se pierda, la RCST cesará de transmitir y reiniciará el procedimiento inicial de sincronización. De igual manera cualquier falla de la RCST durante cualquiera de los siguientes procedimientos descritos llevará a la RCST de nuevo al procedimiento inicial de sincronización.

- La RCST recibirá el TBTP transmitido por el NCC en intervalos regulares. El TBTP está presente en la señalización del *forward link* o FLS y está compuesto por supertramas, tramas y TCTs. La RCST también deberá adquirir la tabla TIM de difusión.

Después de seguir todos estos pasos, la RCST entrará en el estado de Recepción de Sincronización (*Receive Sync*), para proceder a continuación a iniciar el proceso de sincronización.

2.3.5.3 Procedimiento de entrada (*Logon*)

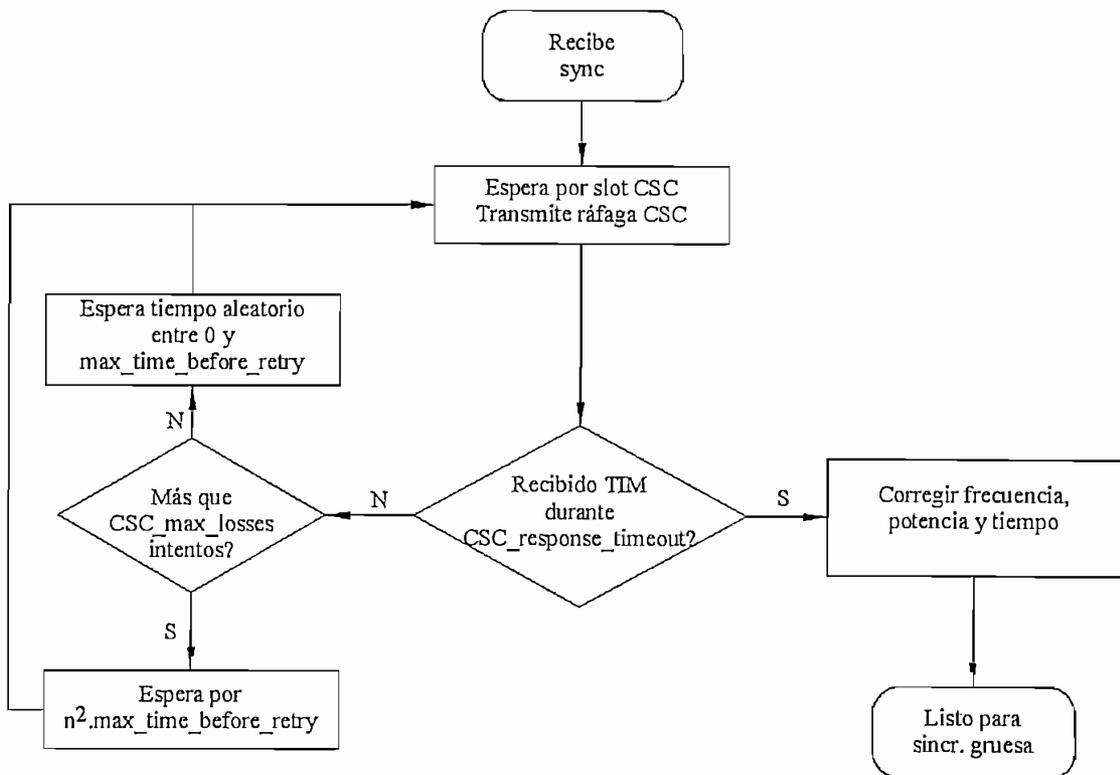


Figura 2.33 Procedimiento de entrada.

Después de que la RCST ha recibido todas las tablas SI relacionadas con la estructura de la Red Satelital Interactiva a través de FLS (ver sección 2.3.8), ésta está lista para inicializar su entrada a la red, con el fin de ser admitida dentro del sistema y enviar tráfico.

La RCST puede decidir dejar el estado *Receive Sync*, por ejemplo, porque está iniciándose o porque desea transmitir datos y no se volverá a registrar en la red después de un largo período de inactividad.

Alternativamente, la red puede activar el procedimiento de entrada enviando una señal denominada *Wake-up* a la RCST. El procedimiento de entrada es ilustrado en el diagrama de flujo de la figura 2.33.

2.3.5.4 Procedimiento de sincronización gruesa (Opcional)

En una red donde todas las RCSTs están sincronizadas al NCR en el enlace de ida, el NCC puede corregir todos los errores de frecuencia y temporización entre la RCST y él mismo.

Después de que la RCST ha entrado a la red y ha recibido autorización para operar, ésta comenzará una fase de adquisición para alcanzar la sincronización de frecuencia y temporización, así como un ajuste de potencia. El procedimiento se ilustra en la figura 2.34.

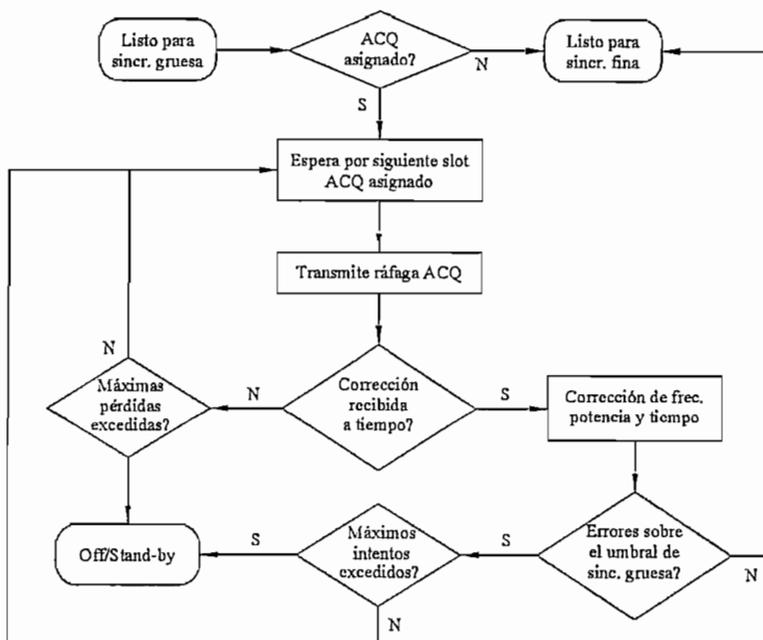


Figura 2.34 Procedimiento de sincronización gruesa.

A través de la TIM se asignan ráfagas ACQ a la RCST. La RCST envía una ráfaga ACQ en su ranura de tiempo reservada. El NCC mide los errores de tiempo, frecuencia y potencia de esta ráfaga ACQ, relativos al sistema de referencia, y envía esta información de vuelta en la CMT (*Correction Message Table*).

La RCST en respuesta, ajusta sus parámetros de transmisión y reintenta. Este proceso continua hasta que el promedio esté dentro de los umbrales que especifica el procedimiento de sincronización gruesa indicado a las RCSTs en la misma ráfaga ACQ.

2.3.5.5 Procedimiento de sincronización fina (Opcional)

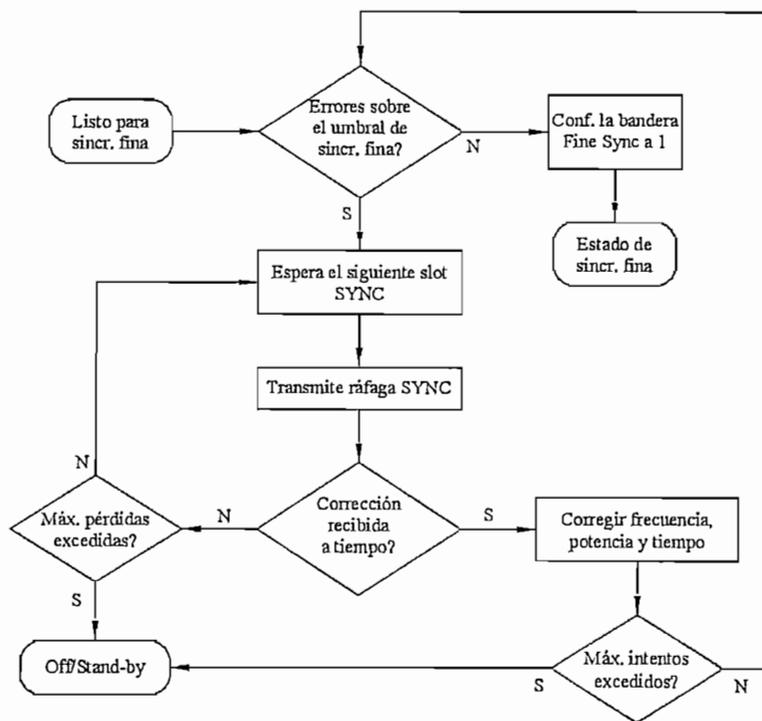


Figura 2.35 Procedimiento de sincronización fina.

Este procedimiento es un tanto similar en principio al Procedimiento de Sincronización Gruesa descrito anteriormente, pero éste usa ranuras SYNC dedicadas en lugar de ráfagas ACQ. El procedimiento de sincronización fina se ilustra en la figura 2.35.

Este procedimiento es únicamente desarrollado si los errores indicados en el último mensaje de corrección (el cual fue incluido en la tabla TIM recibida durante el *TLP* justo después de la CSC, o en la CMT durante el procedimiento opcional ACQ) son más grandes que los umbrales indicados en el procedimiento de sincronización fina presentes en el descriptor SYNC Assign de la TIM recibida en el *TLP*.

2.3.5.6 Procedimiento para mantener la sincronización

Una vez alcanzada la sincronización fina, la RCST está en capacidad de transmitir ráfagas TRF. El procedimiento para mantener la sincronización, se lleva a cabo continuamente durante toda la sesión como se muestra en la figura 2.36.

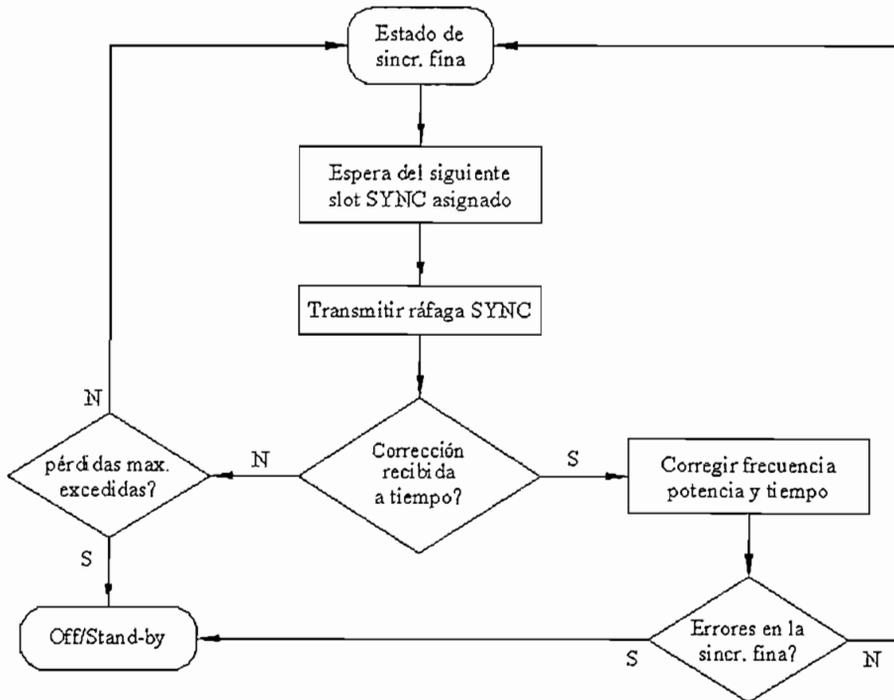


Figura 2.36 Procedimiento para mantener la sincronización.

2.3.5.7 Procedimiento de salida (logoff)

El procedimiento de salida de la RCST descrito a continuación se aplica cuando la RCST se encuentra en el estado de Sincronización Fina (*Fine Sync*).

Cuando la RCST abandona la red, ésta debe cesar la transmisión y su dirección lógica y ranura de tiempo SYNC asignadas deben ser removidas de la lista activa y estar disponibles para otras RCSTs que quieran unirse a la red.

El procedimiento de salida es iniciado como resultado de una sesión terminada (normal) o de una falla (anormal) descrito a continuación:

2.3.5.7.1 *Normal*

Un procedimiento de salida normal puede ser iniciado automática o manualmente por el usuario al final de la sesión.

Para que una RCST inicie el proceso de *logoff*, un mensaje de requerimiento de salida deberá ser transmitido. Para que el NCC inicialice el proceso de salida, el mensaje de salida deberá ser llevado por una TIM dirigida a la RCST.

2.3.5.7.2 *Anormal*

Un proceso de salida anormal se da bajo las condiciones de pérdida de sincronismo, por ejemplo:

- No se ha recibido el NCR por algunos segundos consecutivos.
- La ráfaga CMT de corrección no se ha recibido por algunos SYNC consecutivos.

2.3.6 PILA DE PROTOCOLOS

En el enlace de retorno la pila de protocolos está basada en celdas ATM u opcionalmente en paquetes MPEG2-TS asignados dentro de una ráfaga TDMA. Para la transmisión de datagramas IP, las pilas de protocolos usadas en el enlace de retorno consisten en:

- Enlace de retorno basado en ATM: IP/AAL5/ATM.
- Enlace de retorno MPEG opcional: Encapsulación Multiprotocolo sobre MPEG2-TS.

En el enlace de ida, la pila de protocolos está basada en el estándar DVB/MPEG2-TS. Para la transmisión de datagramas IP, las pilas de protocolos usadas en el enlace de ida consisten en:

- Encapsulación Multiprotocolo sobre MPEG2-TS.
- Opcionalmente IP/AAL5/ATM/MPEG-TS en el modo Tubo de Datos (*Data Piping*) así como para habilitar las comunicaciones directas terminal a terminal, en los sistemas satelitales regenerativos.

Es importante anotar que la norma DVB-RCS especifica dos tipos de RCSTs, dependiendo del tipo de servicios que pueden soportar y de la pila de protocolos que usan. Estos dos tipos son las RCST tipo A y la tipo B.

2.3.6.1 RCST Tipo A (IP)

La RCST tipo A es capaz de soportar únicamente servicios sobre IP, y puede trabajar con celdas ATM o con paquetes MPEG-2. Las celdas ATM son usadas en el enlace de retorno para beneficiarse de la función de segmentación y reensamblaje SAR AAL5, mientras que los paquetes MPEG-2 para aprovechar las ventajas que el encapsulado MPE brinda.

En este tipo de RCST, las celdas ATM pueden ser usadas tanto para la transmisión de tráfico de usuario como para el tráfico de control y administración (manejado por capas superiores). Para este fin se usarán celdas ATM tipo UNI.

Los PIDs para el tráfico de usuario y, si es aplicable, los mensajes MAC a ser usados en los paquetes MPEG2-TS del enlace de retorno se configuran en el

descriptor de inicialización recibido durante el TLP u obtenidos a través de la señalización de la capa superior.

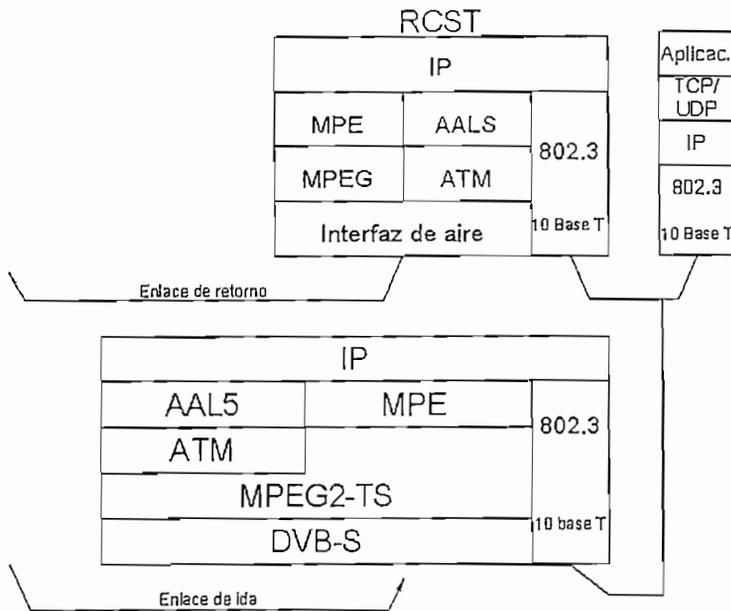


Figura 2.37 Ejemplo de una pila de protocolos para tráfico de usuario con un RCST Tipo A (IP/AAL5/ATM/MPEG2/DVBS es opcional en el enlace de ida).

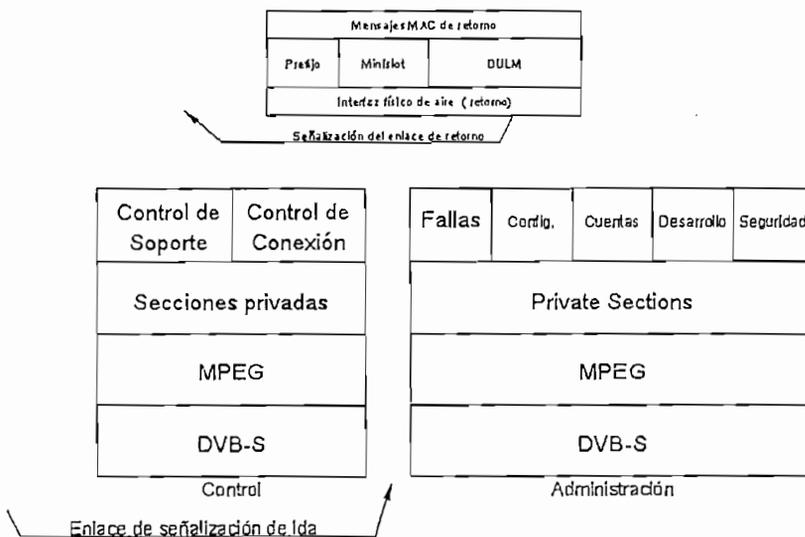


Figura 2.38 Pila de protocolos para señalización.

El tráfico de usuario es enviado entre el equipo del usuario y un *gateway*, u opcionalmente entre dos equipos de usuario, mientras que la señalización es enviada únicamente entre la NIU (*Network Interface Unit*) y el NCC.

La figura 2.37 muestra un ejemplo de una pila de protocolos para tráfico de usuario. La RCST y el equipo de usuario se conectan a través de una red 10BaseT, para el ejemplo. IP se usa para tráfico de usuario en la capa red. La figura 2.38, por su parte, muestra la pila de protocolos para señalización de control y administración.

2.3.6.2 RCST Opcional Tipo B (ATM Nativo)

La RCST Tipo B está contemplada para ser capaz de operar como una RCST del Tipo A y además soportar protocolos nativos ATM encapsulando celdas ATM dentro del MPEG2-TS en el enlace de ida.

En el enlace de retorno las celdas ATM serán asignadas dentro de las ráfagas de tráfico MF-TDMA. Además, la RCST puede soportar PVCs y SVCs en los enlaces de ida y retorno como un UNI.

A diferencia de un ambiente ATM normal existe un medio compartido entre el terminal y el *gateway*. Durante el procedimiento de entrada, un VPI/VCI se asigna a la RCST por el descriptor *logon_initialize* el cual es llevado en la TIM (ver sección 2.3.8.7).

2.3.7 DIRECCIONAMIENTO DE LAS RCSTs

En el *Forward Link* DVB-S, las RCSTs serán únicamente identificados a través de una dirección física MAC y una dirección lógica. La dirección MAC, como ya es conocido, es una dirección física que se encontrará almacenada en

una memoria no volátil presente en el equipo. Ésta corresponde a un identificador físico RSCT único, y sigue las especificaciones del estándar IEEE 802.3.

El valor 0xFFFFFFFF es reservado para una difusión o *broadcast* dirigido a todas las RCSTs de la red. La dirección MAC será usada dentro de las ráfagas CSC y en secciones de datos privadas DSM-CC que llevan las tablas TIM. También se utilizarán para encapsular datagramas IP dentro de tramas MPEG2-TS, tal como se explicó en la sección 2.2.3.5.

La dirección lógica está compuesta de dos campos: *Group_ID* y *Logon_ID*, los cuales son asignados a cada RCST durante el procedimiento de entrada; y se usan para referenciar a una RCST individual (por ejemplo con fines de asignación de capacidad) hasta el procedimiento de *logoff*.

El *Group_ID* corresponde a un grupo de RCSTs admitidas ya en la red, y consiste de un campo de ocho bits, en donde el valor 0xFF se reserva para uso exclusivo del sistema.

El *Logon_ID*, por otra parte, identifica únicamente a una RCST dentro de un grupo identificado con el *Group_ID*. El *Logon_ID* consiste de 16 bits, de donde el valor 0xFFFF se reserva para uso del sistema (modo de contención en el enlace de retorno).

Para una RCST tipo A específica, cualquier dato (tráfico de usuario) que es destinado a ella será transmitido usando la dirección MAC de dicha RCST. Mientras que, cualquier dato que sea destinado a todas las RCSTs tipo A será transmitido usando la dirección MAC de difusión (0xFFFFFFFF).

Cabe señalar que la norma deja abierta la posibilidad de que, independientemente, cada protocolo utilizado en las capas superiores podrá imponer su propio esquema de direccionamiento, por ejemplo el direccionamiento IP.

2.3.8 SEÑALIZACIÓN DEL ENLACE DE IDA O FLS (*Forward Link Signalling*)

Como ya se mencionó en la sección 2.2.2, DVB define un conjunto de tablas construidas sobre la PSI propia de MPEG-2 (ver sección 2.1.4), con el fin de proveer información detallada acerca de la red de difusión DVB-S. Tales tablas DVB son referenciadas como Tablas de Información del Servicio (SI).

2.3.8 SEÑALIZACIÓN DEL ENLACE DE IDA O FLS (*Forward Link Signalling*)

Como ya se mencionó en la sección 2.2.2, DVB define un conjunto de tablas construidas sobre la PSI propia de MPEG-2 (ver sección 2.1.4), con el fin de proveer información detallada acerca de la red de difusión DVB-S. Tales tablas DVB son referenciadas como Tablas de Información del Servicio (SI).

Por otra parte, en una red satelital de dos vías o interactiva, constituida por un enlace de ida (*forward*) y uno de retorno ambos vía satélite, la información de acceso al medio MAC, de control y otros tipos de señalización son comunicados a través del FLS y son transmitidos según el formato DVB-S. Es así que, para la red interactiva DVB-S/RCS, deberán ser aplicadas las especificaciones de la SI de los sistemas DVB en conjunción con algunas tablas añadidas por el estándar RCS.

EL FLS consiste, entonces, de tablas generales SI (las especificadas por DVB-S, MPEG-2 y las propias de DVB-RCS), que transportan información con respecto a la estructura de la red satelital interactiva, mensajes RCST específicos enviados a RCSTs individuales, campos de datos privados definidos por las tablas DVB-SI, paquetes TS especiales (por ejemplo los de inserción del PCR) y algunos descriptores.

Más adelante se revisarán en detalle aquellas tablas que la norma RCS Incluye dentro del DVB-SI, y que son de gran importancia para entender el cómo un terminal RCST obtiene toda la información necesaria de la red para empezar a interactuar con ella. Estas tablas son las siguientes:

- *Superframe Composition Table (SCT)*
- *Frame Composition Table (FCT)*
- *Time-Slot Composition Table (TCT)*
- *Satellite Position Table (SPT)*
- *Correction Message Table (CMT)*
- *Terminal Burst Time Plan (TBTP)*
- *Terminal Information Message (TIM)*

La figura 2.39 ilustra el *stack* de protocolos del FLS, el cual tiene total correspondencia con el mostrado en la figura 2.38. Se puede observar la inserción de las tablas de la norma RCS, MPEG y DVB-S como parte del conjunto de tablas que forman la Información del Servicio del sistema interactivo.

Mensajes específicos RCST (formato de datos privados DSC-CC) Tablas SI Generales (Formato de tabla MPEG2)

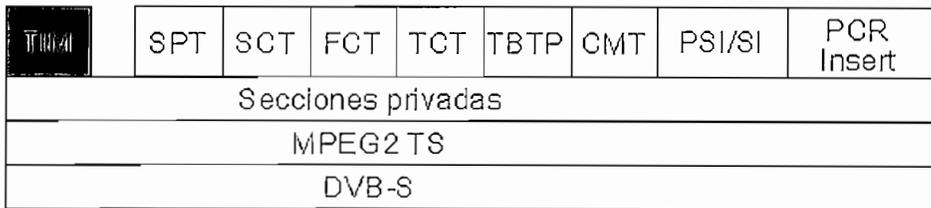


Figura 2.39 Stack de Protocolos para el FLS.

Cabe notar que las tablas SI para las redes satelitales interactivas DVB-S/RCS, son transmitidas en secciones privadas, tal como lo sugiere MPEG-2. Para este fin existen identificadores PID y *Table_id* correspondientes a cada una de las tablas SI, situación que se puede apreciar en la tabla 2.16.

Tablas para la red interactiva	PID	Table_id
RMT	Asignado (ver nota)	0x41
SCT	Asignado	0xA0
FCT	Asignado	0xA1
TCT	Asignado	0xA2
SPT	Asignado	0xA3
CMT	Asignado	0xA4
TBTP	Asignado	0xA5
PCR	Asignado	0xA6
Reservado		0xA7
Reservado		0xA8
Reservado		0xA9
Reservado para uso futuro		0xAB a 0xAF
TIM	Asignado	0xB0
Reservado para uso futuro		0xB1 A 0xBF

NOTA: Este PID deberá ser definido para ser un valor dado dentro de todas las redes interactivas en un sistema dado.

Tabla 2.16 Distribución de los PID y *Table_id* para las tablas SI.

2.3.8.1 *Superframe Composition Table (SCT)*

Esta tabla describe la subdivisión de la totalidad de la red satelital interactiva en supertramas y tramas. La tabla contiene, para cada supertrama, un *superframe_id* (identificador de supertrama), una frecuencia central, un tiempo de inicio absoluto expresado en un valor NCR y un *superframe_count* (contador de supertrama).

Cada supertrama es en lo posterior dividida en tramas, siendo identificado cada tipo de trama con un *frame_id*. La posición de una trama dentro de una supertrama está dada por un "número de trama" usado por las asignaciones de ranuras de tráfico en el TBTP. El posicionamiento de las tramas es relativo a la frecuencia central y al tiempo de inicio de la supertrama asociada.

2.3.8.2 *Frame Composition Table (FCT)*

La FCT describe la partición de las tramas en ranuras de tiempo o *time-slots*. Esta tabla contiene por cada *frame_id* (por ejemplo para cada tipo de trama) una duración de trama, el número total de *time-slots* contenidos en una trama, los tiempos de inicio y los desplazamientos de frecuencia para cada *time-slot*. La transmisión de ciertos parámetros (tales como velocidad de símbolo, velocidad de código, preámbulo, etc) de cada *time-slot* son referenciados por un identificador de *time-slot* transportado por la TCT.

2.3.8.3 *Time-Slot Composition Table (TCT)*

La TCT define los parámetros de transmisión para cada tipo de *time-slot* referenciado por su respectivo identificador. Esta tabla contiene varios campos, los cuales proveen de información a la RCST acerca de las propiedades de los distintos *time-slots*, tales como velocidad de símbolo, velocidad de código, tipo de modulación, preámbulo, contenido del *payload* (TRF con celdas ATM, TRF con paquetes MPEG-2, CSC, ACQ, SYNC) entre otras.

2.3.8.4 *Satellite Position Table (SPT)*

La SPT transporta información acerca de la posición de todos los satélites usados por los enlaces de ida y de retorno. En el caso en el que el NCC sea capaz de calcular el rango RCST-Satélite nominal en períodos de reloj de 27MHz al momento en que el RCST se registra en la red, la SPT no necesita ser transmitida.

2.3.8.5 *Correction Message Table (CMT)*

Esta tabla se utiliza principalmente durante el proceso de *TLP* o entrada de una RCST a la red (aunque puede mantenerse durante toda la sesión), en el cual el NCC envía mensajes CMT a todos los grupos de RCSTs presentes. El propósito de esta tabla es el de informar a la RCST acerca de las correcciones que deben ser hechas a las ráfagas que va a transmitir.

La CMT provee valores de corrección para las ráfagas tales como frecuencia, potencia, temporización y amplitud. Es importante señalar que esta tabla contiene las correcciones para las RCSTs con los valores más recientes obtenidos de las ráfagas ACQ y SYNC.

Es decir que el NCC censa los parámetros de transmisión de la RCST a través de estos últimos *slots*, para luego, en el caso de ser necesario, enviar un mensaje CMT de corrección.

2.3.8.6 *Terminal Burst Time Plan (TBTP)*

El TBTP define la asignación dinámica de *time-slots* generada en cada supertrama. Este mensaje es enviado por el NCC a un grupo de terminales. El grupo es direccionado por un identificador lógico Group_ID, mientras que cada terminal individual es direccionado con un identificador lógico Logon_ID.

Ambos, el *Group_ID* y el *Logon_ID* son notificados al terminal durante el proceso TLP a través del descriptor *Logon_initialize_descriptor* presente en la tabla TIM, a verse más adelante.

Esta tabla contiene una o más entradas para cada RCST, con cada entrada definiendo una asignación de un bloque de *time-slots* contiguos. Cada asignación de tráfico es definida por el número del primer *time-slot* en el bloque y por un factor de repetición que proporciona el número de *time-slots* consecutivos asignados.

Se debe indicar que el TBTP permite que los *time-slots* sean asignados en una sola vez o en forma continua. En este último caso se provee un mecanismo para adicionar o remover asignaciones de *time-slots* dentro del TBTP.

2.3.8.7 Terminal Information Message (TIM)

Este mensaje es enviado por el NCC, ya sea a una RCST individual direccionado por su dirección MAC (mensaje *unicast*), o a todas las RCSTs a través de un *broadcast* con la dirección MAC de *broadcast* reservada; y contiene información estática o cuasi-estática acerca del enlace de ida (*forward link*) tal como su configuración.

La TIM también puede ser usada para facilitar el cambio de una RCST hacia un diferente grupo de red, o para cambiar un grupo de RCSTs a un servicio FLS de otro MPEG-2 TS por ejemplo.

Dentro de esta tabla se encuentran una variedad de descriptores, los cuales proveen información adicional a la RCST, pero quizá el de mayor importancia es el *Logon_initialize_descriptor*. Este descriptor le suministra a la RCST algunos parámetros necesarios para el proceso inicial TLP.

Parámetros tales como el *Group_ID* y *Logon_ID*, son indicados por este descriptor. Como ha sido mencionado ya varias veces el *Group_ID* y *Logon_ID*

son dos identificadores que permiten referenciar de forma única a una RCST durante los procesos de acceso MAC, señalización, *TLP*, etc.

El *Logon_initialize_descriptor* indica también aspectos tales como si los métodos DULM, de mini ranuras, mini ranuras en contención, etc, son soportados por el sistema. De esta forma la RCST puede tener un conjunto de información suficiente, a través de la TIM, como para poder acoplarse a las características de la red.

2.3.9 ACCESO AL *FORWARD LINK SIGNALLING* (FLS)

La Información del Servicio para el *Forward Link Signalling* (FLS) puede ser transmitida en uno o más servicios FLS. Al ser DVB un servicio de datos, el FLS debe ser referenciado como cualquier otro servicio o programa dentro de la *Program Association Table* (PAT), *Program Map Table* (PMT) y opcionalmente dentro de la *Service Description Table* (SDT), todas ellas son tablas MPEG-2 y SI vistas en la sección 2.1.4 y 2.2.2 respectivamente.

La PMT del servicio FLS contiene los PIDs correspondientes al NCR, a las tablas generales SI y a los mensajes específicos RCST tal como fueron definidos en la tabla 2.16.

Existe información adicional acerca de la Red Satelital Interactiva que es transportada por una tabla denominada *RCS Map Table* (RMT). Esta tabla que tiene la misma sintaxis que la NIT (pero es transmitida con distinto PID) contiene uno o más descriptores que apuntan a uno a más servicios FLS.

Cada descriptor contiene a su vez, una lista de *population_id*, los cuales son usados por la RCST para seleccionar su FLS apropiado. Con este propósito la RCST tiene un *population_id* previamente configurado por defecto en el proceso de instalación que puede ser cambiado y grabado por el sistema para procesos TLP posteriores. Los *population_id* son únicos en toda la red.

El procedimiento de Adquisición del *Forward Link* es concerniente con la adquisición del enlace de ida (*forward*) que transporta el servicio FLS. Este procedimiento comprende los siguientes pasos:

- Sincronización física del enlace
- Localización del enlace de ida
- Sincronización del NCR
- Carga de las Tablas DVB-RCS específicas

2.3.9.1 Sincronización física del enlace

La sincronización física del enlace consiste en el proceso de ajuste o sintonización del *transponder* del satélite, recuperación de la señal transmitida y sincronización al formato del flujo de transporte DVB de dicha señal.

Este procedimiento necesita llevarse a cabo cada vez que una RCST requiera moverse a un *transponder* diferente.

2.3.9.2 Localización del enlace de ida

El procedimiento de localización del enlace de ida consiste en identificar al *transponder* que lleva al servicio FLS y los PIDs correspondientes. Este procedimiento utiliza tres tablas DVB estándares, las cuales son:

- *Network Information Table* (NIT)
- *Program Association Table* (PAT)
- *Program Map Table* (PMT)

Los pasos que se siguen para la localización del FLS son ilustrados en la figura 2.40. Se debe señalar que este procedimiento necesita que los siguientes parámetros sean previamente configurados en la RCST como valores por defecto:

- Detalles de la ubicación del TS inicial o primario (*start-up*). Éste puede ser cualquier TS en la red, y es aquel en el que se encuentra multiplexada la información de señalización RCS.
- El identificador *population_ID*, que le permite a la RCST formar parte de un grupo de estaciones cuyo TS es único dentro de la red.

- Detalles de la ubicación del TS inicial o primario (*start-up*). Éste puede ser cualquier TS en la red, y es aquel en el que se encuentra multiplexada la información de señalización RCS.
- El identificador *population_ID*, que le permite a la RCST formar parte de un grupo de estaciones cuyo TS es único dentro de la red.

La RCST deberá sintonizar el TS inicial y buscar por una copia de la NIT para ubicar el TS que contiene la tabla RMT. Esta última será identificada por la localización del descriptor *linkcage_type* = 0x07 que es el correspondiente para ella.

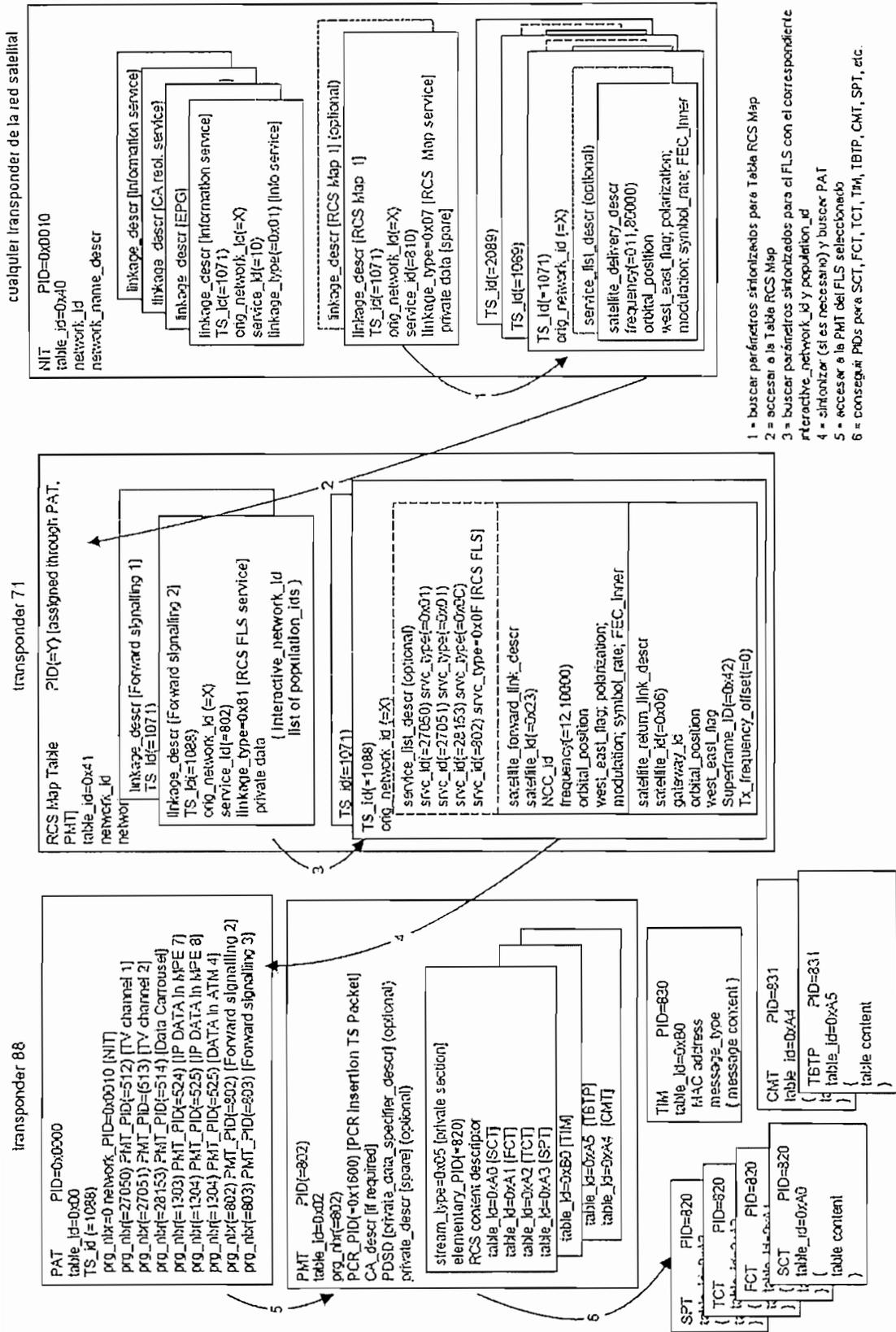
Una vez identificado el TS al cual pertenece la RMT, la RCTS deberá copiar sus características y configurar su receptor a los parámetros de dicho flujo con el fin de ubicar la RMT (ver figura 2.40).

Con la RMT encontrada y copiada, el terminal buscará el TS asociado a su *population_ID*, ya que éste le indicará los PIDs correspondientes a las tablas de Información de Servicio DVB-S/RCS características de su flujo de transporte.

El paso siguiente será el hallar la PAT e identificar en ella los PIDs de las PMTs asociadas a cada servicio presente en el múltiplex de transporte, para que de esta forma la RCST obtenga toda la información necesaria de las distintas tablas del servicio FLS y pueda formar parte de la red interactiva.

2.3.9.3 Sincronización del NCR

Éste es un procedimiento establecido por la norma para fijar el temporizador de la RCST al contador NCR provisto por el NCC. Esto se logra a través de la recepción de los paquetes MPEG-2 TS que contienen el PCR (*Program Clock Reference*, referirse a la sección 2.1.3.2.1).



- 1 - buscar parámetros sincronizados para Tabla RCS Map
- 2 - acceder a la Tabla RCS Map
- 3 - buscar parámetros sincronizados para el FLS con el correspondiente interactive_network_id y population_id
- 4 - sincronizar (si es necesario) y buscar PAT
- 5 - acceder a la PMT del FLS seleccionado
- 6 - conseguir PDS para SCT, FCT, TCT, TIM, TBTP, CMT, SPT, etc.

Figura 2.40 Ejemplo de Localización del enlace de ida.

Una vez alcanzado un correcto nivel de recuperación del NCR, la RCST deberá seguir monitoreándolo. Como medida de seguridad y prevención la norma establece que si la recepción del NCR y del servicio FLS es interrumpida, la RCST deberá cesar la transmisión con el fin de no afectar al sistema.

2.3.9.4 Carga de las tablas DVB-RCS específicas

La RCST tendrá que cargar las tablas que se especifican a continuación, usando los PIDs identificados en el procedimiento de localización del enlace de ida:

- SCT, FCT y TCT para identificar las estructuras disponibles de tramas y supertramas.
- SPT para obtener la información de la ubicación del satélite.
- TIM (versión *broadcast*) para identificar posibles anomalías del enlace.

Además, la RCST deberá estar alerta en busca de posibles mensajes TIM *unicast* tipo *wake-up* direccionados con la dirección MAC del terminal, los cuales son llamados a la RCST para que empiece a interactuar con la red.

2.3.10 SEÑALIZACIÓN DEL CANAL DE RETORNO

Es ya conocido que el NCC maneja la sincronización de una RCST a través de un proceso que consiste en una serie de mensajes de señalización intercambiados entre ambos sobre *time-slots* reservados para este fin (SYNC, CSYNC, CSC, ACQ), lo que le permite al terminal alcanzar una sintonización fina de todos los parámetros de sincronización, temporización, frecuencia y potencia.

Con el fin de medir los parámetros de transmisión de la RCST, para luego corregirlos si es el caso, el NCC le pide que envíe una secuencia de ráfagas ACQ asignadas a través del TBTP para un número limitado de repeticiones.

También existe la posibilidad de que en sistemas que usan la transmisión periódica de ráfagas SYNC, el NCC a su vez requiera de la RCST una transmisión periódica de ráfagas SYNC vía la TIM a través de la asignación de *slots* SYNC cada *n* supertramas.

Los mensajes de corrección son enviados a través de las tablas CMT para que la RCST ajuste de mejor manera sus parámetros de transmisión. A continuación se presenta el esquema de envío de señalización de la red interactiva DVB-S/RCS.

Enlace de Ida: Mensaje/DSM-CC y Sección SI/MPEG-2 TS/(DVB-S)

Enlace de Retorno: Ráfagas Especiales/Interfaz aire

Mensajes usados: TIM (*forward*) – [DSM-CC] o TBTP [SI]
CMT (*forward*) – [SI]
CSC (retorno)
ACQ (retorno)
SYNC (retorno)

El NCC usa los parámetros de configuración intercambiados entre las RCSTs y él para identificar la capacidad funcional del terminal, y de esa manera determinar qué características de transmisión puede demandar de una RCST en particular.

En adición, el NCC puede obtener información tal como identificación del fabricante, versión de la RCST (número de receptores del FLS), tipo de RCST (potencia de amplificación), identificación de usuario, versión de *hardware*, versión del *software*, posición de la RCST (latitud, longitud, altitud), características de la ODU (potencia, tamaño y ganancia de la antena), tipo de conexión RCST (SMATV, usuario simple, múltiples usuarios), identificación del instalador, código postal, fecha y hora de instalación, etc.

Referencias Bibliográficas del Capítulo 2

- [1] A. Delgado, "Flujos de Programa y de Transporte MPEG-2 Aplicación A DVB" Universidad Politécnica de Madrid, Junio 2001
- [2] ISO/IEC, "*Generic Coding of Moving Pictures and Associated Audio*", Recomendación H.222.0. ISO/IEC 13818-1, 1994.
- [3] ETSI Standard, "*Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation, for 11/12 GHz satellite services*", EN 300 421, 1997
- [4] ETSI Technical Report 101 211 v1.6:1, "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)", May 2004.
- [5] ETSI Technical Report 101 202, "*Digital Video Broadcasting (DVB); Implementation Guidelines for Data Broadcasting*", v1.1.1 February 1999
- [6] ETSI Standard, "*Digital Video Broadcasting (DVB); DVB specification for data broadcasting*", EN 301 192 v1.3.1, May 2003
- [7] ETSI Standard, "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems", EN 301 790 v1.4.1, April 2005.
- [8] H. Clausen, H. Linder, B. Collin-Nocker, "*Internet over Direct Broadcast Satellites*", IEEE Communications Magazine, June 1999.

CAPÍTULO 3

NETWORK SIMULATOR

Como se plantea en el objetivo general del presente proyecto, el propósito primordial del mismo es la simulación de una red VSAT DVB-S/RCS. Para ello, se debe contar con una herramienta de simulación que sea confiable y que produzca los datos necesarios para claramente evaluar el comportamiento de la red.

El *software* escogido para la simulación es un sistema que durante varios años ha sido muy utilizado en el entorno mundial ya que gracias a su estructura flexible y modular, permite añadir funcionalidades de tal manera que su desempeño sea óptimo y eficiente.

En el presente capítulo se expone de manera resumida las principales características y funcionalidades del programa de simulación de redes *Network Simulator* (NS). Los temas a tratarse aquí, ayudarán al lector en lo posterior, a entender de mejor manera cómo la simulación de la red VSAT se implementa en NS.

3.1 PRINCIPIOS DEL NS

NS es un simulador de redes manejado por eventos, que, permite simular una gran variedad de redes IP. NS fue desarrollado por el proyecto VINT (*Virtual InterNetwork Testbed*), el cual a su vez es un esfuerzo conjunto de la UC at Berkeley, USC-ISI, LBL, y Xerox PARC. El proyecto está apoyado por la Agencia de Proyectos de Investigación para Defensa Avanzada (DARPA).

NS implementa protocolos de red tales como TCP y UDP; comportamiento de fuentes de tráfico como FTP, Telnet, Web, CBR y VBR; mecanismos de manejo de colas como *Droptail*, RED y CBQ; algoritmos de ruteo como Dijkstra, etc. NS también implementa protocolos *multicasting* y algunos de los protocolos de la capa MAC para simular LANs [1].

Como se observa en la figura 3.1, NS es un intérprete de *scripts* del lenguaje Tcl Orientado a Objetos, el cual tiene un planificador de eventos de simulación, librerías de objetos componentes de red y librerías de módulos de instalación de red (implementación). En otras palabras, para crear simulaciones en NS, se debe programar en lenguaje de *scripts* Otcl.

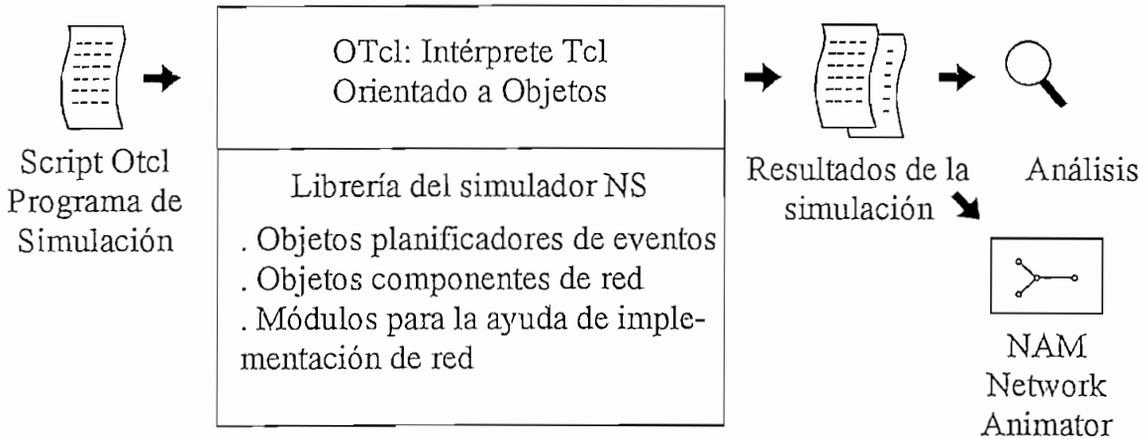


Figura 3.1 Esquema de simulación en NS. [1]

Para establecer y ejecutar una simulación de red, el usuario debe escribir un *script* Otcl que inicialice un planificador de eventos, establezca la topología de la red usando las funciones de las librerías mencionadas, y diga a las fuentes de tráfico cuándo empezar y cuándo dejar de transmitir paquetes a través del planificador de eventos.

El término implementar se usa para describir la configuración de la red, esto es básicamente describir las posibles trayectorias entre objetos, configurando los punteros vecinos de un objeto a la dirección de otro apropiado.

Cuando el usuario quiere crear un nuevo objeto de red, lo puede hacer ya sea escribiendo uno nuevo o creando un objeto compuesto de las librerías existentes, e implementar el camino de los datos a través de dicho objeto.

Un componente importante del NS, aparte de los objetos de red, es el planificador de eventos o *event scheduler*. Un evento en NS es un identificador de

paquete que es único para el mismo con un tiempo planificado y un apuntador al objeto que maneja el evento. El planificador de eventos entonces guarda la pista del tiempo de simulación y ejecuta todos los eventos programados en sus respectivos momentos al invocar los componentes de red apropiados, y los deja realizar la acción asociada con el paquete apuntado por el evento.

Los componentes de la red se comunican entre sí al pasarse los paquetes; sin embargo, esto no consume tiempo de simulación real. Todos los componentes de red que necesitan gastar cierto tiempo de simulación manipulando un paquete (un retardo por ejemplo), usan el planificador de eventos atribuyendo un evento al paquete y esperando a que éste sea disparado por sí mismo antes de hacer cualquier acción de manipulación del paquete.

Por ejemplo, un componente de conmutación de red que simula un conmutador con 20 us de latencia, produce un evento para un paquete a ser conmutado al planificador como un evento 20 us después. El planificador después de 20 us, desencolará el evento y ejecutará el componente de conmutación, el cual a su vez pasará el paquete hacia un enlace de salida apropiado.

Otro uso para el planificador de eventos es el de temporizador. Por ejemplo, TCP necesita un temporizador para mantener el control del tiempo de transmisión de un paquete para su retransmisión (transmisión de un paquete con el mismo número de paquete TCP, pero con diferente ID de paquete para NS).

Los temporizadores utilizan el planificador de eventos de manera similar que los retardos; la única diferencia es que los temporizadores miden un valor de tiempo asociado con un paquete y hacen una acción apropiada referida a dicho paquete después de que pase un cierto tiempo, y no simulan un retardo.

NS está escrito, no sólo en OTcl sino también en C++. Por razones de eficiencia, NS separa la implementación de trayectoria de datos de las implementaciones de las trayectorias de control. Con el fin de reducir el tiempo de

proceso de paquetes y eventos (tiempo de no simulación), el planificador de eventos y los objetos de componentes básicos de red en la trayectoria de datos están escritos y compilados en C++ [1].

Estos objetos compilados se hacen disponibles al intérprete OTcl a través de un acoplamiento OTcl, el cual crea un objeto OTcl compatible para cada uno de los objetos C++, y hace que las funciones de control y las variables configurables (especificadas por el objeto C++) actúen como funciones miembro y variables miembro del correspondiente objeto OTcl. De esta misma manera, los controles del objeto C++, se dan a OTcl.

También es posible agregar funciones y variables miembro a un objeto OTcl enlazado con C++. Los objetos C++ que no necesitan ser controlados en una simulación o internamente ser utilizados por otro objeto, no necesitan ser enlazados a OTcl. Asimismo, un objeto (que no está en la trayectoria de datos) puede ser totalmente implementado en OTcl.

La figura 3.2, muestra un ejemplo de la jerarquía de objetos en C++ y OTcl. Una cosa que se debe notar en la figura es que para cada objeto C++ que tiene un enlace con OTcl (formando una jerarquía), existe una jerarquía de objeto OTcl correspondiente y muy similar.

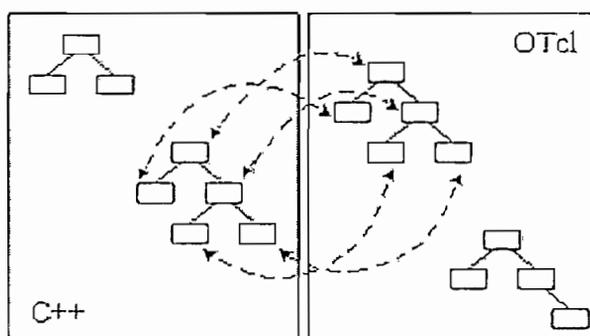


Figura 3.2 Dualidad C++ y Otcl. [1]

La figura 3.3, muestra la arquitectura general del *Network Simulator*. Se puede apreciar que un usuario general (que no es un diseñador NS) se puede

considerar ubicado en la esquina inferior izquierda, ejecutando las simulaciones en Tcl, usando objetos del simulador existentes en las librerías OTcl.

Los planificadores de eventos y la mayoría de los componentes de red están implementados en C++ y a su vez están disponibles para OTcl a través de un enlace OTcl que es implementado usando Tclcl (Tcl Class). Todos estos elementos en conjunto, forman el *Network Simulator*, el cual es un intérprete Tcl orientado a objetos con librerías de simulación de redes.

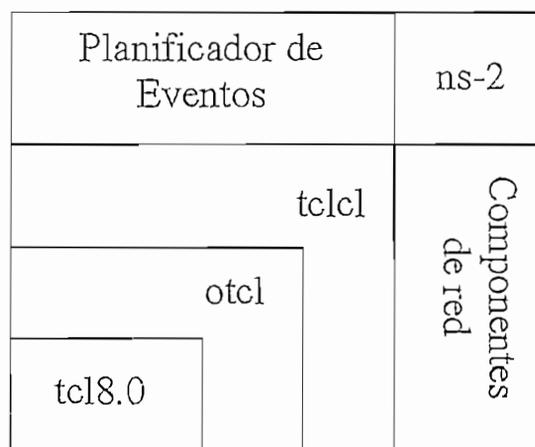


Figure 3.3 Arquitectura de NS. [1]

Un aspecto importante en NS es la obtención de resultados de las simulaciones. Como se indica en la figura 3.1, cuando una simulación termina, NS produce uno o más archivos de texto de salida que contienen datos detallados de la simulación, siempre y cuando en el *script* OTcl se lo haya especificado.

Los datos pueden ser usados para analizar la simulación o para desplegar una simulación gráfica con la herramienta *Network Animator* (NAM) que también fue desarrollada por el proyecto VINT y que será descrita en lo posterior.

Para ejecutar los ejemplos planteados en las secciones posteriores, es necesario que el *software* NS se encuentre correctamente instalado bajo Linux.¹

¹ La instalación de NS se detalla en el Anexo A.

3.1.1 LENGUAJE DE USUARIO OTCL

Como se mencionó anteriormente, NS es básicamente un intérprete OTcl con librerías de objetos de simulación de red. Es muy útil conocer como programar en OTcl para usar NS.

A continuación se muestra un ejemplo de un *script* Tcl y OTcl, desde el cual se puede tener una idea más clara de la programación en OTcl.

```
# Procedimiento llamado "test"
proc test {} {
    set a 43
    set b 27
    set c [expr $a + $b]
    set d [expr [expr $a - $b] * $c]
    for {set k 0} {$k < 7} {incr k} {
        if {$k < 5} {
            puts "k < 5, pow = [expr pow($d, $k)]"
        } else {
            puts "k > 5, mod = [expr $d % $k]"
        }
    }
}

# Llamada al procedimiento "test"
test
```

Ejemplo 3.1 *Script* Tcl sencillo.

El ejemplo 3.1 es un *script* general Tcl que muestra cómo crear y llamar un procedimiento, cómo asignar valores a variables y cómo implementar un lazo. Sabiendo que OTcl es una extensión orientada a objetos de Tcl, es obvio que todos los comandos Tcl trabajan en OTcl. La relación entre Tcl y OTcl es la misma que existe entre C y C++.

Para ejecutar el *script* del ejemplo 3.1 primeramente se llamará a éste con el nombre de archivo "ex-tcl.tcl" seguidamente se escribirá en la línea de comandos "ns ex-tcl.tcl" (el comando "ns" inicia el NS (intérprete OTcl)). Se

consigue el mismo resultado escribiendo “tcl ex-tcl.tcl”, si tcl8.0 está instalado en la máquina.

En Tcl, el comando **proc** es usado para definir un procedimiento, seguido de un nombre de procedimiento y los argumentos encerrados entre llaves { }. El comando **set** es usado para asignar valores a una variable. [**expr ...**] se usa para que el intérprete calcule el valor de la expresión que sigue a la palabra “expr”. Para obtener el valor asignado a una variable se utiliza **\$** con el nombre de la variable. El comando **puts** imprime en pantalla la cadena de caracteres entre las comillas “ ”. A continuación se muestra el resultado de ejecutar el ejemplo 3.1.

```
k < 5, pow = 1.0
k < 5, pow = 1120.0
k < 5, pow = 1254400.0
k < 5, pow = 1404928000.0
k < 5, pow = 1573519360000.0
k >= 5, mod = 0
k >= 5, mod = 4
```

El siguiente es un ejemplo de la programación orientada a objetos en OTcl. Muestra la manera cómo un objeto es creado y utilizado en OTcl.

Como un usuario NS ordinario, las posibilidades de escribir por sí mismo un objeto propietario, son escasas. Sin embargo, todos los objetos NS que se usen en un programa de simulación, escritos o no en C++ y disponibles en OTcl a través del enlace, o escritos únicamente en OTcl, son esencialmente objetos OTcl.

El Ejemplo 3.2, es un *script* OTcl que define dos clases de objetos, “mom” y “kid”, donde “kid” es una subclase (o clase *child*) de “mom”, y una función miembro llamada “saludo” para cada clase. Después de las definiciones de clase, se declara cada objeto; la variable “edad_” de cada instancia está inicializada en 45 (para “mom”) y 15 (para “kid”), y después se llama a la función miembro “saludo” de cada objeto.

```
# Inclusión de una función miembro llamada "saludo"
Class mom
mom instproc saludo {} {
    $self instvar edad_
    puts "Mamá de $edad_ años dice:
    ¿Qué haces?"
}

# Creación de una clase child de "mom" llamada "kid" y
# sustitución de la función miembro "saludo"
Class kid -superclass mom
kid instproc saludo {} {
    $self instvar edad_
    puts "Hijo de $edad_ años dice:
    Hola Mamá"
}

# Creación de un objeto mom y uno kid, configurando cada edad
set a [new mom]
$a set edad_ 45
set b [new kid]
$b set edad_ 15

# Llamada de la función miembro "saludo" de cada objeto
$a saludo
$b saludo
```

Ejemplo 3.2 Script OTcl sencillo. [1]

El comando **Class** se usa para crear una clase objeto e **instproc** para definir una función miembro a una clase objeto. La herencia de clase está especificada al usar el comando **-superclass**. En la definición de las funciones miembro, **\$self** actúa de la misma forma que el puntero "this" en C++ [1], e **instvar** chequea si el nombre de la variable contigua está ya declarada en su clase o en su superclase. Si el nombre de la variable dada está ya declarada, la variable es referenciada, si no, se declara una nueva. Finalmente, para crear un objeto, el comando **new** se usa como se indica en el ejemplo.

Llamando al ejemplo como "ex-otcl.tcl" y ejecutando "ns ex-otcl.tcl", se obtendrá el siguiente resultado:

```
Mamá de 45 años dice:
¿Qué haces?
Hijo de 15 años dice:
Hola Mamá
```

3.1.2 EJEMPLO DE SIMULACIÓN SENCILLO

El ejemplo 3.3, es un *script* OTcl que crea una configuración simple de red y ejecuta el escenario de simulación.

Esta red consiste de 4 nodos (n0, n1, n2, n3) como se muestra en la figura 3.4. Los enlaces *duplex* entre n0 y n2, y entre n1 y n2 tienen una capacidad de 2 Mbps y un retardo de 10 ms. El enlace *duplex* entre n2 y n3 tiene una capacidad de 1.7 Mbps y un retardo de 20 ms.

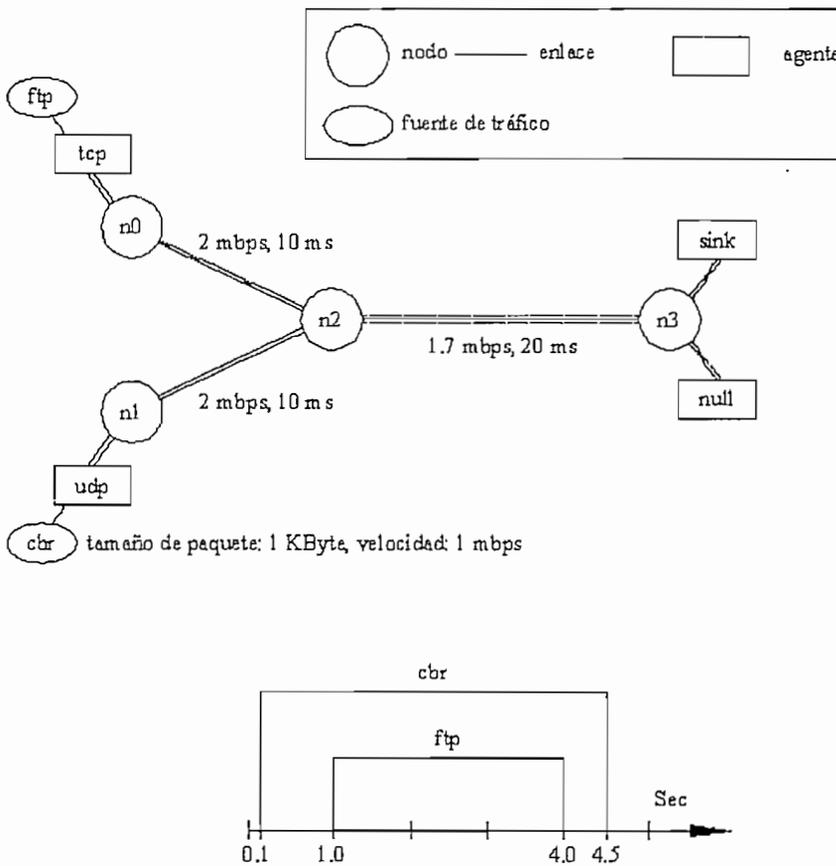


Figura 3.4 Una topología de red simple y un escenario de simulación. [1]

Cada nodo usa una cola *DropTail*, cuyo tamaño máximo es 10. El nodo n0 tiene adjunto un agente "tcp", y el nodo n3 establece una conexión tcp con el agente "sink". Por omisión, el tamaño máximo de un paquete que un agente "tcp" genera es 1 KByte. Un agente tcp "sink" genera y envía paquetes ACK al emisor (agente tcp) y libera los paquetes recibidos.

Un agente "udp" adjunto al nodo n1 está conectado a un agente "null" adjunto al nodo n3. Un agente "null" a diferencia del "sink", únicamente libera los paquetes recibidos.

Generadores de tráfico "ftp" y "cbr" están adjuntos a los agentes "tcp" y "udp" respectivamente; "cbr" está configurado para generar paquetes de 1 KByte con una velocidad de 1 Mbps. El generador "cbr" está configurado para iniciar a los 0.1 s y terminar a los 4.5 s, y el operador "ftp" para iniciar a 1.0 s y terminar a los 4.0 s.

```
# Crea un objeto de simulación
set ns [new Simulator]

# Define diferentes colores para el flujo de datos (para el NAM)
$ns color 1 Blue
$ns color 2 Red

# Abre el archivo de trazo NAM
set nf [open out.nam w]
$ns namtrace-all $nf

# Define el procedimiento 'finish'
proc finish {} {
    global ns nf
    $ns flush-trace
    # Cierra el archivo de trazo NAM
    close $nf
    # Ejecuta NAM en el archivo de trazo
    exec nam out.nam &
    exit 0
}

# Crea cuatro nodos
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# Crea enlaces entre los nodos
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

# Establece el tamaño de cola del enlace n2-n3 en 10
$ns queue-limit $n2 $n3 10

# Posiciona los nodos (para NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
```

```
# Controla la cola para el enlace n2-n3 (para NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

# Configura una conexión TCP
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

# Configura FTP sobre una conexión TCP
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

# Configura una conexión UDP
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

# Configura CBR sobre una conexión UDP
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

# Planifica eventos para los agents CBR y FTP
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

# Retira los agents tcp y sink (no necesario)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3
$sink"

# Llama al procedimiento 'finish' después de 5 segundos de
simulación
$ns at 5.0 "finish"

# Imprime el tamaño e intervalo de los paquetes CBR
puts "CBR tamaño de paquete = [$cbr set packet_size_]"
puts "CBR intervalo = [$cbr set interval_]"

# Ejecuta la simulación
$ns run
```

Ejemplo 3.3 *Script de simulación NS.*

En general, un *script* NS inicia creando un objeto Simulator [2].

- **set ns [new Simulator]:** genera un objeto de simulación NS, y lo asigna a la variable *ns* (en esta sección se utilizará letra itálica para diferenciar las variables). Lo que hace este comando es lo siguiente:
 - Inicializa el formato de paquete (se analizará posteriormente)
 - Crea el planificador
 - Selecciona el formato de dirección por omisión (se analizará posteriormente)

El objeto "Simulator" entre otras, tiene las siguientes funciones miembro:

- Crear objetos compuestos tales como nodos y enlaces (descritos después)
- Conectar componentes de red con objetos creados (por ejemplo adjuntar un agente)
- Configurar los parámetros de los componentes de red (en su mayoría objetos compuestos)
- Crear conexiones entre agentes (por ejemplo crear conexiones entre "tcp" y "sink")
- Especificar opciones para mostrar en el NAM

Muchas de las funciones miembro son para la configuración de la simulación y el planeamiento, sin embargo algunas de ellas sirven para configurar la visualización en el NAM. Las implementaciones de la función miembro del objeto "Simulator" se localizan en el archivo "~ns/tcl/lib/ns-lib.tcl".

- ***\$ns color fid color:*** se usa para establecer el color de los paquetes para un flujo específico a través de su identificador de flujo (*fid*). Esta función miembro del objeto "Simulator" es para la visualización en el NAM, y para el presente efecto no tiene relevancia.

- **`$ns namtrace-all nombre_de_archivo`**: Esta función miembro le dice al simulador que almacene los trazos de simulación en un formato de entrada NAM. También da el nombre del archivo en el cual los trazos serán escritos más adelante con el comando **`$ns flush-trace`**. De igual manera, la función miembro **`trace-all`** almacena los trazos de simulación en un formato general.
- **`proc finish {}`**: es llamado, cuando se desee que esta simulación termine, con el comando **`$ns at 5.0 "finish"`**. En esta función se especifican los procesos post-simulación.
- **`set n0 [$ns node]`**: La función miembro **`node`** crea un nodo. Un nodo en NS es un objeto compuesto hecho de clasificadores de dirección y puerto (descritos más adelante). Los usuarios pueden crear un nodo separadamente, creando un objeto clasificador de dirección y uno de puerto y, conectándolos entre sí. Sin embargo, esta función miembro del objeto Simulator hace el trabajo más sencillo.
- **`$ns duplex-link nodo1 nodo2 capacidad retardo tipo_de_cola`**: crea dos enlaces *simplex* de capacidad y retardo especificados, y conecta los nodos en cuestión. En NS, la cola de salida de un nodo está implementada como parte de un enlace, es por eso que los usuarios deberían especificar el tipo de cola cuando se creen los enlaces. En el ejemplo 3.3 se usa una cola *DropTail*. Si se desea usar una cola RED (*Random Early Discard*), simplemente se reemplaza la palabra *DropTail* por RED. Al igual que un nodo, un enlace es un objeto compuesto, y se puede crear sus sub-objetos y conectarlos a los nodos.
- **`$ns queue-limit nodo1 nodo2 número`**: Esta línea configura el límite de cola de los dos enlaces *simplex* que conectan a los nodos *nodo1* y *nodo2* a un número específico.
- **`$ns duplex-link-op nodo1 nodo2 ...`**: usado para visualización en el NAM.

Ahora que la configuración básica de la red está hecha, lo siguiente es configurar los agentes de tráfico tal como TCP y UDP, fuentes de tráfico tal como FTP y CBR, y adjuntarlos a los nodos y agentes respectivamente [2].

- **set tcp [new Agent/TCP]:** Esta línea muestra cómo crear un agente TCP. Pero en general, los usuarios pueden crear cualquier agente o fuente de tráfico de esta manera. Los agentes y fuentes de tráfico son de hecho objetos básicos (no objetos compuestos), mayormente implementados en C++ y enlazados a OTcl. Es por eso que no existe una función miembro del objeto Simulator que cree estos objetos. Para crear agentes o fuentes de tráfico, un usuario debería conocer los nombres de las clases de estos objetos (Agent/TCP, Agent/TCPSink, Application/FTP, etc.). Esta información se la puede encontrar en el manual del NS, o en el archivo “~ns/tcl/libs/ns-default.tcl”
- **\$ns attach-agent nodo agente:** La función miembro **attach-agent** adjunta un objeto agente ya creado a un objeto nodo.
- **\$ns connect agente1 agente2:** Esta línea, después de que dos agentes que se comunicarán uno al otro han sido creados, establece una conexión de red lógica entre ellos.

Asumiendo que la configuración de red está hecha, lo siguiente es escribir un escenario de simulación (por ejemplo simulación planificada). El objeto Simulator tiene muchas funciones miembro de planificación. Sin embargo, la más usada es la siguiente:

- **\$ns at tiempo “cadena_de_comandos”:** Esta función miembro del objeto Simulator hace que el planificador ejecute la cadena de comandos a un tiempo de simulación dado. Por ejemplo, **\$ns at 0.1 “\$cbr start”** hará que el planificador llame a la función miembro **start** del objeto de fuente de tráfico CBR, con lo cual inicia el CBR para transmitir datos. En NS, comúnmente una fuente de tráfico no transmite datos reales, pero notifica al agente fundamental

que tiene cierta cantidad de datos para transmitir, y el agente crea paquetes conociendo cuántos de los datos va a transmitir y los envía.

Una vez que se haya configurado la red y que las especificaciones de los procedimientos de post-simulación y planificación estén hechas, se ejecuta la simulación con el comando `$ns run`.

3.1.3 COMPONENTES DE RED

La figura 3.5 muestra parcialmente una jerarquía de clases OTcl de NS, las cuales ayudarán a entender los componentes básicos de una red.

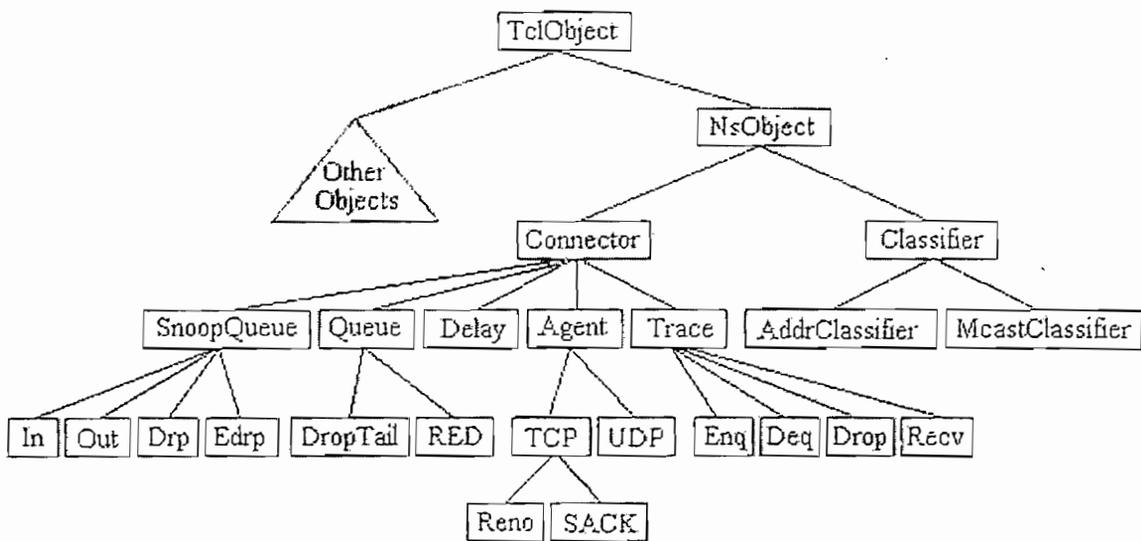


Figura 3.5 Jerarquía de clases (parcial). [1]

La raíz de la jerarquía es la clase `TclObject` que es la superclase de todos los objetos de librería OTcl (planificadores, componentes de red, temporizadores y los otros objetos incluyendo los relacionados con el NAM). Como una clase progenitora de `TclObject`, la clase `NsObject` es la superclase de todos los objetos componentes básicos de red que manipulan paquetes, los cuales están conformados por objetos de red compuestos tales como nodos y enlaces.

Los componentes de red básicos están divididos en dos subclases, *Connector* y *Classifier*, basados en el número de posibles rutas de salida de datos. Los objetos de red básicos que tienen una única ruta de salida de datos se encuentran bajo la clase *Connector*, y los objetos conmutados que tienen múltiples posibilidades de ruta para la salida de datos están bajo la clase *Classifier*.

3.1.3.1 Nodos y ruteo

Un nodo es un objeto compuesto de un objeto de entrada de nodo y clasificadores como se muestra en la figura 3.6.

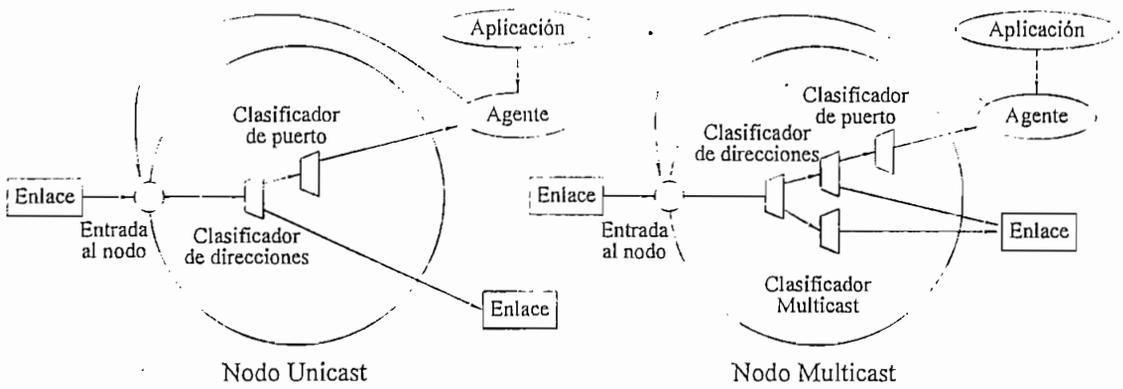


Figura 3.6 Nodo (*Unicast* y *Multicast*). [1]

Existen dos tipos de nodos en NS. Un nodo *unicast* tiene un clasificador de dirección, que hace un ruteo *unicast*, y un clasificador de puerto. Un nodo *multicast*, además de lo presente en el *unicast*, tiene un clasificador de paquetes *multicast* y *unicast*, y un clasificador *multicast* que desempeña el ruteo *multicast*.

En NS, los nodos *unicast* están dados por defecto. Para crear nodos *multicast* el usuario debe notificar explícitamente en el *script* OTcl, después de crear un objeto de planificación, que todos los nodos a ser creados serán nodos

multicast. Después de especificar el tipo de nodo, el usuario también puede seleccionar un protocolo de ruteo específico diferente al usado por omisión.

- Unicast
 - *\$ns rproto type*
 - *type: Static, Session, DV, cost, multi-path*

- Multicast
 - *\$ns multicast*
 - *\$ns mrtproto type*
 - *type: CtrMcast, DM, ST, BST*

3.1.3.2 Enlaces

Un enlace es otro objeto compuesto en NS. Cuando un usuario crea un enlace usando la función miembro “duplex-link” de un objeto Simulator, se crean dos enlaces *simples* en ambas direcciones tal como se muestra en la figura 3.7.

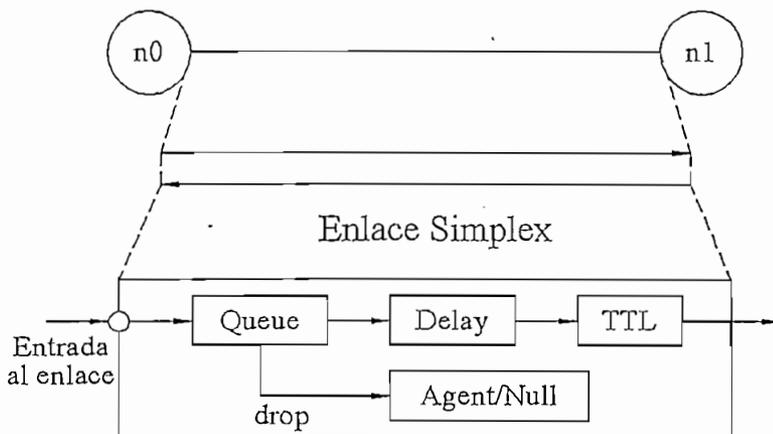


Figura 3.7 Enlace. [1]

Cabe recalcar que una cola de salida de un nodo es realmente implementada como parte del objeto enlace *simplex*. Los paquetes desencolados

desde la cola se pasan al objeto *Delay* que simula el retardo del enlace, y los paquetes desechados en una cola se envían a un agente nulo y son liberados ahí. Finalmente, el objeto TTL calcula los parámetros de tiempo de vida (*Time To Live*) para cada paquete recibido y actualiza el campo TTL del paquete.

3.1.3.2.1 Trazo

En NS, las actividades de red son trazadas alrededor de los enlaces *simplex*. Si el simulador está directamente trazando las actividades de red (especificado por `$ns trace-all archivo` o `$ns namtrace-all archivo`), los enlaces creados después del comando tendrán los objetos de trazo insertados que se muestra en la figura 3.8. Los usuarios pueden también específicamente crear un objeto de trazo del tipo *type* entre los nodos dados *src* y *dst* usando el comando `create-trace {tipo archivo src dst}` [2].

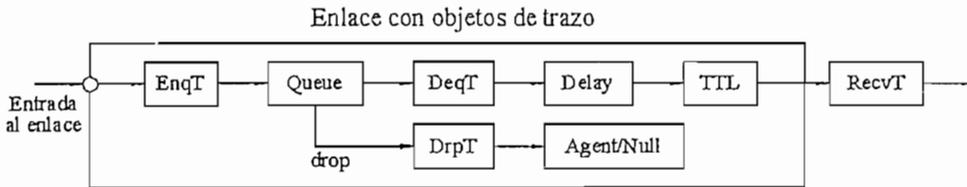


Figura 3.8 Objetos de trazo insertados.

Cuando cada objeto de trazo insertado (*EnqT*, *DeqT*, *DropT* y *RecvT*) recibe un paquete, escribe al archivo de trazo especificado sin consumir tiempo de simulación, y pasa el paquete al siguiente objeto de red.

EnqT registra la información acerca de un paquete que llega y se encola en la cola de entrada del enlace. Si el paquete se desborda, la información de éste es manejada por *DropT*.

DeqT registra la información en el momento en que el paquete es desencolado. Finalmente, *RecvT* da información sobre paquetes que se han recibido en la salida del enlace.

3.1.3.2.2 Control de cola

Básicamente el trazo de objetos está diseñado para grabar el tiempo de llegada de los paquetes en el cual ellos están programados. Aunque un usuario consiga suficiente información del trazo, podría estar interesado en conocer qué está pasando dentro de una cola de salida específica.

El control de cola se puede alcanzar utilizando objetos de control de cola y objetos "curiosos" de cola como se muestra en la figura 3.9.

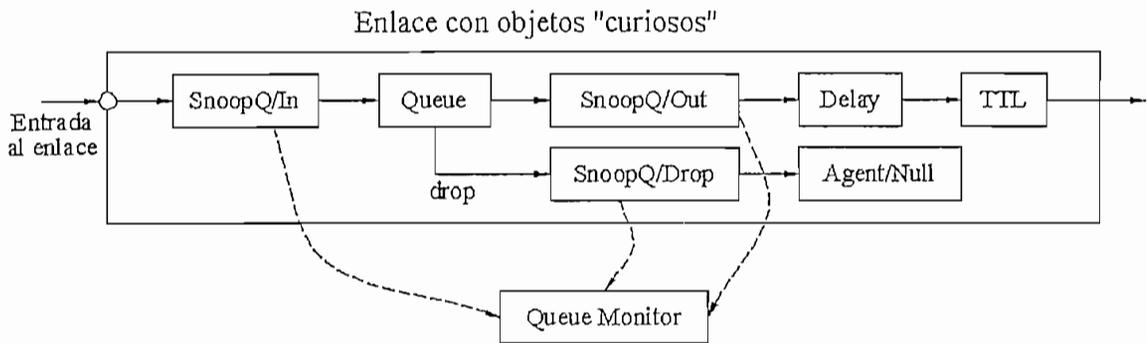


Figura 3.9 Control de cola. [1]

Cuando un paquete llega, un objeto "curioso" de cola notifica al objeto de control de cola acerca de este evento. Se debe notar que los objetos "curiosos" de cola se usan en paralelo con los objetos de trazo aún cuando no se muestran en la figura 3.9.

3.1.3.3 Ejemplo de flujo de paquete

Hasta ahora se han examinado los dos componentes de red más importantes (nodos y enlaces). La figura 3.10 muestra el interior de una configuración de simulación de red y del flujo de paquete.

La red consiste de dos nodos (n_0 y n_1) cuyas direcciones de red son 0 y 1 respectivamente. Un agente TCP está implementado en n_0 y un agente *Sink* en n_1 , ambos en el puerto 0, de esta manera TCP se comunica con el sumidero *Sink*

a través de dicho puerto. Finalmente, una aplicación FTP (o fuente de tráfico) está implementada sobre el agente TCP, solicitando enviar cierta cantidad de datos.

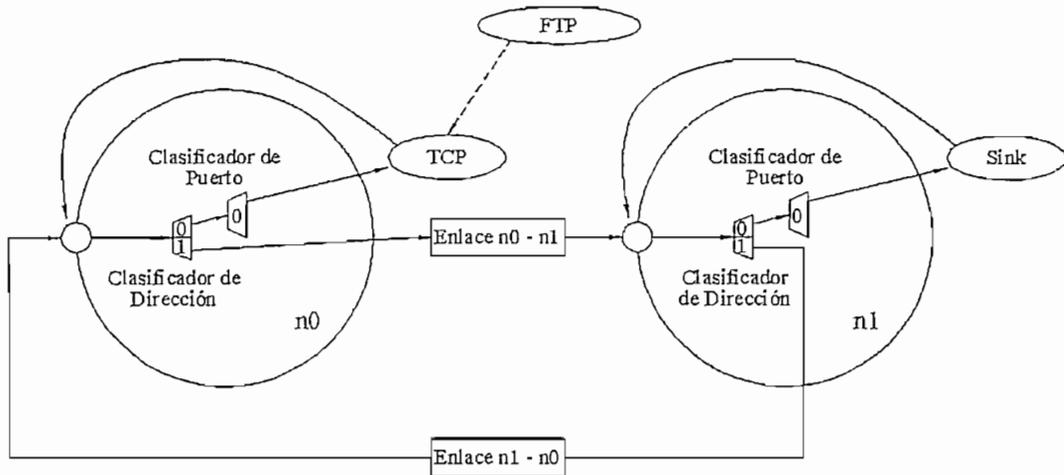


Figura 3.10 Ejemplo de flujo de paquete. [1]

Note que la figura 3.10 no indica el comportamiento exacto de una aplicación FTP sobre TCP, únicamente muestra el interior detallado de la configuración de simulación de red y un flujo de paquete.

3.1.4 PAQUETE

Un paquete NS está compuesto de una pila de cabeceras y de un espacio opcional de datos (figura 3.11). Como ya se mencionó brevemente, se inicializa el formato de cabecera de paquete cuando se crea un objeto Simulator, donde se define una pila de todas las cabeceras registradas, tal como la cabecera común que es mayormente utilizada por cualquier objeto como cabecera IP, TCP, RTP (UDP usa RTP) y de trazo, y también se almacena el desplazamiento de cada cabecera en la pila [2].

Esto significa que, se use o no una cabecera específica, una pila compuesta de todas las cabeceras registradas se crea cuando un paquete se

coloca por un agente, y un objeto de red puede acceder a cualquier cabecera en la pila de un paquete usando el valor del desplazamiento correspondiente.

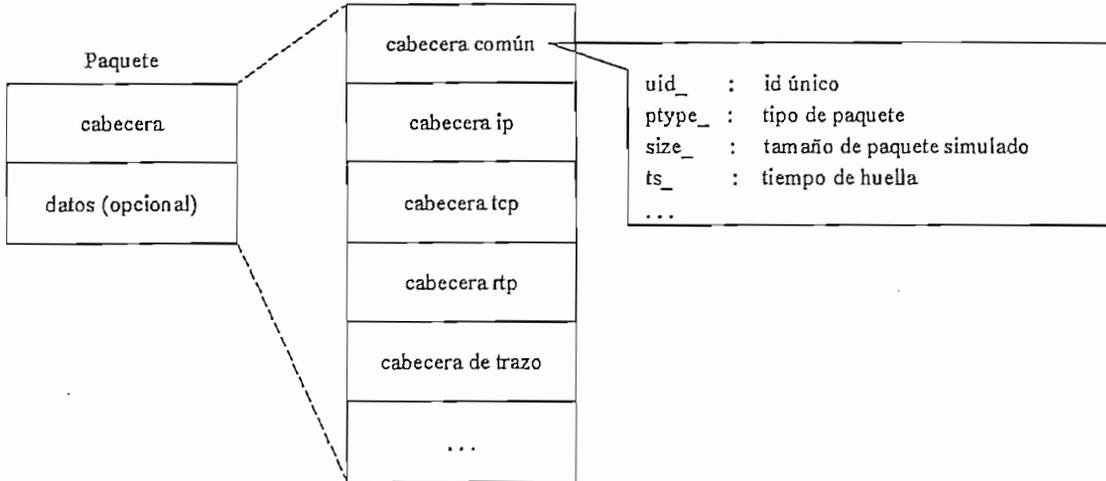


Figura 3.11 Formato de un paquete NS. [1]

Muchas veces un paquete tiene la cabecera de pila y un apuntador de espacio de datos que es nulo. A pesar de que un paquete puede llevar datos reales (desde una aplicación) colocando un espacio de datos, muy pocas aplicaciones e implementaciones de agentes lo soportan. Esto se debe a que es insignificante llevar datos en una simulación de tiempo no real.

Sin embargo, si se desea implementar esta característica en donde una aplicación transmita datos a otra a través de la red, se la puede utilizar con una pequeña modificación en la implementación del agente fundamental.

Otra aproximación posible sería crear una nueva cabecera para la aplicación y modificar el agente fundamental para escribir los datos recibidos desde la aplicación a la nueva cabecera.

3.1.5 ARCHIVOS DE TRAZO

El ejemplo 3.4 es el mismo *script* OTcl del ejemplo 3.3 añadido comandos (en letra negra), que crean un archivo out.tr para registrar los trazos que NS genera en las simulaciones.

```
...
# Abre el archivo de trazo NAM
set nf [open out.nam w]
$ns namtrace-all $nf

# Abre el archivo de trazo
set tf [open out.tr w]
$ns trace-all $tf

# Define el procedimiento 'finish'
proc finish {} {
    global ns nf
    $ns flush-trace
    # Cierra el archivo de trazo NAM
    close $nf
    # Cierra el archivo de trazo
    close $tf
    # Ejecuta NAM en el archive de trazo
    exec nam out.nam &
    exit 0
}
...
```

Ejemplo 3.4 Ejemplo 3.3 habilitado el archivo de trazo. [1]

Ejecutando el *script* anterior se genera un archivo de trazo NAM, que será usado por NAM, y un archivo de trazo llamado "out.tr" que se usará para analizar la simulación.

La figura 3.12 muestra el formato del archivo "out.tr" generado en el ejemplo 3.4.

La estructura del trazo contiene los siguientes campos [3]:

a) El primer campo es el tipo de evento. Está dado por uno de cuatro posibles símbolos r, +, -, d que corresponden respectivamente a recibir (a la salida del enlace), encolado, desencolado y caído.

evento	tiempo	nodo origen	nodo destino	tipo paquete	tamaño paquete	banderas	fid	dirección origen	dirección destino	número seq.	id de paquete
--------	--------	-------------	--------------	--------------	----------------	----------	-----	------------------	-------------------	-------------	---------------

r : recibido (en el nodo destino)
 + : desencolado (en la cola)
 - : desencolado (en la cola)
 d : rechazado (en la cola)

dirección origen : nodo.puerto (3.0)
 dirección destino : nodo.puerto (0.0)

```
r 1.3556 3 2 ack 40 ----- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ----- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ----- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ----- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ----- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ----- 1 0.0 3.0 29 199
```

Figura 3.12 Ejemplo de formato de trazo.

- b) El segundo campo indica el instante en el cual el evento ocurre.
- c) Nodo de entrada del enlace en el cual el evento ocurre.
- d) Nodo de salida del enlace en el cual el evento ocurre.
- e) Tipo de paquete (por ejemplo, CBR, o TCP).
- f) Tamaño del paquete.
- g) Banderas.
- h) Identificación de flujo (fid) de IPv6 que un usuario puede configurar para cada flujo en la entrada del programa OTcl. Se puede usar este campo además para propósito de análisis; también es usado para especificar el color del flujo que se despliega en el NAM.
- i) Dirección de la fuente dada en el formato "nodo.puerto".
- j) Dirección del destino, dada por el mismo formato.

k) Número de secuencia de paquete del protocolo de la capa red. Aunque las aplicaciones de UDP en una red real no usan el número de secuencia, el NS guarda el número de secuencia de los paquetes UDP para propósito de análisis.

l) Identificación única de paquete.

3.2 MANIPULACIÓN DE LOS ARCHIVOS DE TRAZO [3]

El NS proporciona datos detallados de eventos que ocurren en la red. Si se desea analizar los datos se puede extraer información pertinente de los trazos y manipularlos.

Se puede escribir los programas en cualquier lenguaje de programación que pueda manejar archivos de datos.

Existen varias herramientas para el análisis que están disponibles gratuitamente bajo varios sistemas operativos (Unix, Linux, Windows, etc.). Todas ellas requieren escribir programas cortos que son interpretados y ejecutados sin necesidad de compilarlos.

3.2.1 PROCESAMIENTO DE ARCHIVOS CON AWK

La herramienta awk permite hacer operaciones simples en archivos de datos; tal como, promediar valores de una columna dada, sumar o multiplicar término a término entre varias columnas, etc.

En los siguientes dos ejemplos se mostrará cómo tomar el valor promedio de una columna dada en un archivo, y así calcular la desviación estándar.

```
BEGIN {FS = "\\t"} {n1++} {s=s+$4} END {print "Promedio:" s/n1}
```

Ejemplo 3.5 Programa awk para promediar los valores de la columna 4 de un archivo (promedio.awk).

Se debe usar " " en lugar de "\t" cuando la separación entre columnas es un espacio de carro y no una tabulación.

```
BEGIN {FS="\t"}{ln++}{d=$4-t}{s2=s2+d*d} END {print "Desviación estándar:" sqrt(s2/ln)}
```

Ejemplo 3.6 Programa awk para obtener la desviación estándar de la columna 4 de un archivo (desstd.awk).

Para ejecutar el ejemplo 3.5, y calcular el promedio de la cuarta columna de un archivo llamado "out.ns" se debe teclear en Unix:

```
awk -f promedio.awk out.ns
```

Se obtendrá como resultado:

```
Promedio: 29.397
```

Para calcular la desviación estándar de esa columna, se debe teclear en Unix:

```
awk -v t=29.397 -f desstd.awk out.ns
```

Se obtendrá como resultado:

```
Desviación estándar: 33.2003
```

Nótese que para calcular la desviación estándar se requiere el valor promedio, el cual se obtuvo del ejemplo 3.5, y que al ejecutar desstd.awk, éste es parte del argumento de entrada.

Se puede observar al final del ejemplo 3.5 que, si no se divide para n1, se obtendrá simplemente la suma de entradas de la columna 4 en lugar de su promedio.

Una mejor manera de obtener el promedio y la desviación estándar es a través de vectores, tal como se muestra en el ejemplo 3.7.

```
BEGIN { FS = "\t" } { val [nl]=$4 } {nl++ } {s=s+$4} END {
  av=s/nl
  for (i in val) {
    d=val [i] -av
    s2=s2+d*d
  }
  print "Promedio: " av " Desviación estándar: " sqrt(s2/nl)}
```

Ejemplo 3.7 Programa awk para promedio y desviación estándar.

Los siguientes ejemplos toman como entrada un archivo de 15 columnas (0 a 14). Crea como salida 5 columnas donde la primera contiene la columna no. 1 del archivo original, y las columnas de la 2 a la 5 son la suma de las columnas 3-4, 6-8, 9-11 y 12-14, respectivamente (12-14 corresponden a las tres últimas columnas en el archivo original).

```
BEGIN {FS="\t"}{l1=$3+$4+$5}{l2=$6+$7+$8}{d1=$9+$10+$ 11) \
{d2=$12+$13+$14}{print $1 $1"\t" l1"\t" l2"\t" d1"\t" d2 } END {}
```

Ejemplo 3.8 Programa awk para mover columnas.

La ejecución del ejemplo 3.8 se lo puede hacer tecleando en Unix:

```
awk -f suma.awk Conn4.tr > outfile
```

El archivo original aquí es Conn4.tr y el de salida está escrito en un archivo llamado *outfile*.

3.2.2 USO DE GREP

El comando grep en Unix permite "filtrar" un archivo. Se puede crear un archivo nuevo que consista de solo aquellas líneas del archivo original que contengan una secuencia de caracteres dada.

Por ejemplo, el archivo de salida en NS puede contener todos los tipos de paquetes que pasan por todos los enlaces, y puede ser que se esté interesado solamente en aquellos datos acerca de paquetes tcp que fueron del nodo 0 al nodo 2. Si las líneas que involucran tales eventos contienen la cadena "0 2 tcp" entonces se hará lo siguiente:

```
grep "0 2 tcp" tr1.tr > tr2.tr
```

donde "tr1.tr" es el archivo de trazo original y "tr2.tr" es el nuevo archivo.

Si se desea obtener un archivo que contenga todas las líneas de tr1.tr que empieza con la letra r, se deberá teclear:

```
grep "^r" tr1.tr > tr2.tr
```

Si se desea hacer un archivo de todas las líneas que empiezan con "s" y tengan después "tcp 1020", se debe teclear:

```
grep "^s" simple.tr | grep "tcp 1020" > tr3.tr
```

3.2.3 PROCESAMIENTO DE ARCHIVOS DE DATOS CON PERL

PERL son las siglas de "*Practical Extraction and Report Language*". PERL permite un filtrado y procesamiento fácil de archivos de datos ASCII en el Unix. Este lenguaje fue creado por Larry Wall con la idea principal de simplificar las tareas de administración del sistema.

PERL ha evolucionado mucho y hoy en día es un lenguaje de propósito general y una de las herramientas más utilizadas para una administración de datos de Internet.

```
# teclee: perl throughput.pl <archivo de trazo> <nodo
# requerido> <granularidad> > archivo

$infile=$ARGV[0];
$tonode=$ARGV[1];
$granularity=$ARGV[2];

# Se calculará cuántos bytes fueron transmitidos durante el
# intervalo de tiempo especificado por el parámetro de
# granularidad en segundos.
$sum=0;
$clock=0;

    open (DATA,"<$infile")
        || die "No se puede abrir Sinfile $!";

    while (<DATA>) {
        @x=split(' ');

#Columna 1 es el tiempo
if [$x[1]-$clock <= $granularity)
{
#se chequea si el evento corresponde a un recepción
if ($x[0] eq 'r')
{
#se chequea si el destino corresponde al primer argumento
if ($x[3] eq $tonode)
{
#se chequea si el tipo de paquete es TCP
if ($x[4] eq 'tcp')
{
    $sum=$sum+$x[5];
}
}
}
}
else
{
    $throughput=$sum/$granularity;
    print STDOUT "$x[1] $throughput\n";
    $clock=$clock+$granularity;
    $sum=0;
}
}

    $throughput=$sum/$granularity;
    print STDOUT "$x[1] $throughput\n";
    $clock=$clock+$granularity;
    $sum=0;

    close DATA;
exit(0);
```

Ejemplo 3.9 Programa PERL para calcular la velocidad efectiva.

PERL es un lenguaje interpretado que tiene muchos usos, pero principalmente está dirigido a la búsqueda, extracto y reportes. Algunas ventajas de usar PERL son:

- Fácil implementación de pequeños programas a ser utilizados como filtros, para extraer información de archivos de texto.
- Puede usarse en muchos sistemas operativos sin cambiar el código.
- El mantenimiento y depuración de programas PERL es más simple que en otros lenguajes específicos.
- Es muy popular.

Se presenta en esta sección algunos programas de utilidad.

El ejemplo 3.9 calcula dinámicamente la velocidad efectiva de las conexiones TCP. Promedia la velocidad efectiva sobre períodos definidos por un parámetro llamado granularidad (*granularity*).

Como entrada toma tres argumentos: el nombre de un archivo de trazo (por ejemplo out.tr), el nodo en el que se desea verificar la velocidad efectiva de TCP y la granularidad.

3.2.4 GRAFICACIÓN CON GNUPLOT

Gnuplot es un software gratuito extensamente disponible para ambos sistemas Unix/Linux así como para Windows.

La manera más simple de usar gnuplot es teclear "plot < fn >", dónde el archivo (cuyo nombre se ha escrito como fn) tiene dos columnas que representan los valores **x**, **y** de los puntos.

Los puntos pueden ser unidos por una línea de estilos diferentes escribiendo comandos como:

```
plot 'fn' w lines l
```

(números diferentes pueden darse en lugar de "1", lo que produce estilos de línea diferentes).

Alternativamente, se puede usar diferentes tipos de puntos escribiendo comandos como:

```
plot 'fn' w points 9
```

(de nuevo, se puede pintar varios tipos de puntos dependiendo del número después de la palabra "points").

Algunas otras características de gnuplot se pueden observar en el siguiente ejemplo:

```
set size 0.6,0.6
set pointsize 3
set key 100,8
set xrange [90.0:120.0]
plot 'fn1' w lines l, 'fn2' w lines 8, 'fn3' w points 9
```

- La línea 1 producirá una curva de tamaño menor que la predeterminada.
- La línea 2 producirá puntos que son más grandes que los predeterminados. (En ambas líneas se pueden usar otros números).
- La línea 4 restringe el rango del eje x en el intervalo 90-120.
- La línea 5 sobrepone tres curvas en una sola figura, obtenida de tres archivos diferentes: fn1, fn2 y fn3.
- La línea 3 dice al gnuplot dónde exactamente poner la simbología que definirá el lugar donde irá la leyenda en la figura para describir los objetos trazados. En particular, se da para cada objeto graficado el tipo de línea o de punto que se usará. En lugar de una posición exacta, se podría usar las palabras claves 'left', 'right', 'top', 'bottom', 'outside' y 'below', por ejemplo:

```
set key below
```

(lo cual configura la simbología bajo el gráfico), o simplemente "set nokey" que desactiva la simbología completamente.

El nombre predefinido de cada objeto que aparece en la simbología simplemente es su nombre de archivo correspondiente. Si se desea dar a un objeto un título distinto al nombre de archivo se tiene que especificar en el comando "plot", por ejemplo:

```
plot 'fn1' t "expectation" w lines 1, 'fn2' t "variance" w lines 2
```

Aquí, los nombres "expectation" y "variance" aparecerán en la simbología.

Si las mismas secuencias de comandos son empleadas algunas veces, se puede escribirlas en un archivo, por ejemplo de nombre "g1.com", y entonces simplemente cargar el archivo cada vez que se desee usarlo:

```
load 'g1.com'
```

Gnuplot se puede usar para extraer alguna columna de un archivo multicolumna. Esto se realiza de la siguiente manera:

```
plot 'queue.tr' using 1:($4/1000) t "kbytes" w lines 1, \
'queue.tr' using 1:5 t "packets" w lines 2
```

lo cual grafica primeramente una curva que usa la columna 1 del archivo "queue.tr" como el eje de las **x** y como el de las **y**, los datos de la columna 4 divididos para 1000; después traza en el mismo gráfico usando el mismo eje **x** con la columna 1 y en el eje **y** los datos de la columna 5. Cabe recalcar que el orden entre "using", "t" y "lines" es importante.

En la figura 3.13 se tiene un ejemplo de uso de Gnuplot con el siguiente código:

```
plot 'out.tr' using 2:($6/1000) t "Kbytes" w lines 1, 'out.tr'
using 2:8 t "Paquetes" w lines 2
```

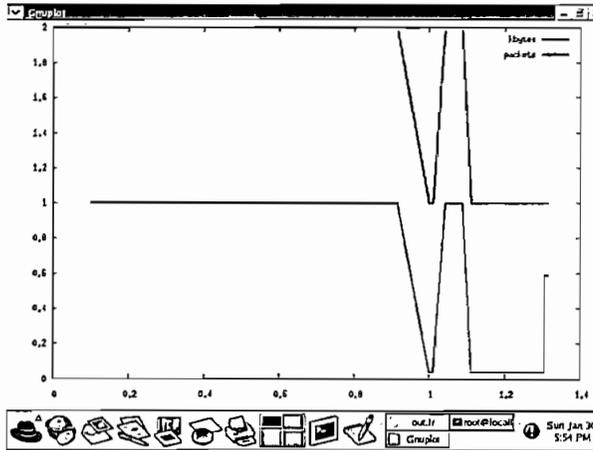


Figura 3.13 Ejemplo de gnuplot.

3.2.5 GRAFICACIÓN CON XGRAPH

Xgraph es una herramienta de graficación proporcionada por el NS.

Xgraph permite crear archivos *postscript*, Tgif, y otros, haciendo clic en el botón "Hdcp". Puede ser invocado dentro de los comandos tcl para desplegar inmediatamente después de que la simulación haya concluido.

Como entradas, el comando xgraph espera uno o más archivos ASCII que contengan datos de pares ordenados x-y en cada línea. Por ejemplo, `xgraph f1 f2` imprimirá en la misma figura los archivos f1 y f2.

Algunas opciones en xgraph son:

- Título: -t "título".
- Tamaño: -geometry xsize x ysize.
- Títulos de ejes: -x "xtitle" (para el título del eje x) y -y "ytitle" (para el título del eje y).
- Color de texto y grilla: con la bandera -v.

Un ejemplo de un comando sería:

```
xgraph f1 f2 -geometry 800x400 -t "Loss rates" -x "time" -y "Lost packets"
```

Un ejemplo de uso del Xgraph, se indica en la figura 3.14, de acuerdo al siguiente código:

```
xgraph out.tr -geometry 800x400 -t "Ejemplo de uso del Xgraph" -x "tiempo" -y "Paquetes perdidos"
```

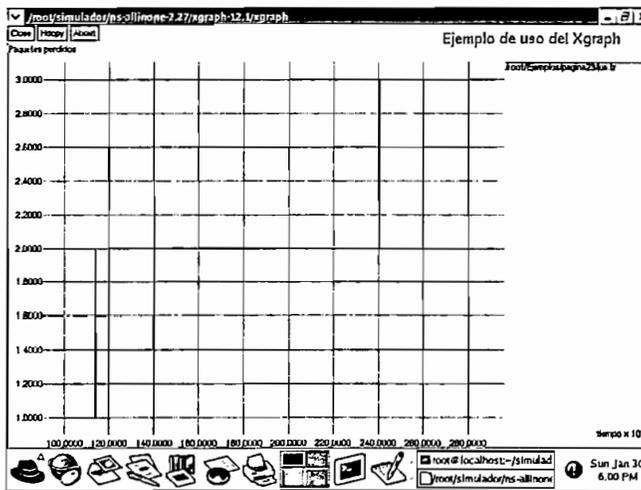


Figura 3.14 Ejemplo de uso del XGraph.

3.3 VARIABLES ALEATORIAS [3]

En NS es posible generar variables aleatorias (RVs) con diferentes distribuciones.

3.3.1 SEMILLAS Y GENERADORES

Además de su distribución, hay otros aspectos que necesitan ser involucrados cuando se simula una variable aleatoria:

- Obtener el mismo valor de una variable aleatoria cuando se ejecute la simulación, posiblemente variando algunos parámetros, permitiría

comparar directamente, para un grupo aleatorio de eventos, cómo los resultados simulados dependen de algunos parámetros físicos (tal como retardo en los enlaces, o longitudes de cola).

- A menudo se necesita variables aleatorias que sean independientes unas de otras.

La generación de variables aleatorias usa una semilla (qué es algún número que se escribe en el programa tcl). El valor 0 de la semilla resulta en la generación de una nueva variable aleatoria cada vez que se ejecuta la simulación; así, si se desea tener las mismas variables aleatorias generadas en simulaciones diferentes, éstas deberían ser almacenadas. En contraste, si se usa otras semillas, cada vez que se ejecute la simulación, se generará la misma secuencia de variables aleatorias que son generadas en una simulación.

En NS, si se usan generadores diferentes con la misma semilla y la misma distribución, ellos crearán los mismos valores de variables aleatorias (a menos que la semilla sea cero). Lo mencionado se verá en un ejemplo más adelante.

3.3.2 CREACIÓN DE VARIABLES ALEATORIAS

Primeramente se crea un nuevo generador y se asigna a él una semilla, por ejemplo el valor 2, con el comando:

```
set MyRng1 [new RNG]
$MyRng2 seed 2
```

Entonces al crear una variable aleatoria, se tiene que definir su tipo de distribución y sus parámetros. A continuación se dan varios ejemplos para crear variables aleatorias con distribuciones de Pareto, Constante, Uniforme, Exponencial e HyperExponencial.

1. **Distribución de Pareto.** Una RV con distribución de Pareto, por ejemplo r_1 , se construye especificando su valor esperado y su forma, parámetro β . Los valores predefinidos son 1.0 y 1.5 respectivamente.

```
set r1 [new RandomVariable/Pareto]
$r1 use-rng $MyRng
$r1 set avg_ 10.0
$r1 set shape_ 1.2
```

2. **Distribución Constante.** Se define de la siguiente manera:

```
set r2 [new RandomVariable/Constant]
$r2 use-rng $MyRng
$r2 set val_ 5.0
```

3. **Distribución uniforme.** Está definida a través del punto más pequeño y el más grande de rango, así:

```
set r3 [new RandomVariable/Uniform]
$r3 use-rng $MyRng
$r3 set min_ 0.0
$r3 set max_ 10.0
```

4. **Distribución exponencial.** Está definida a través de su valor medio, así:

```
set r4 [new RandomVariable/Exponential]
$r4 use-rng $MyRng
$r4 set avg_ 5
```

5. **Distribución Hyperexponencial.** Está definida de la manera siguiente:

```
set r5 [new RandomVariable/HyperExponential]
$r5 use-rng $MyRng
$r5 set avg_ 1.0
$r5 set cov_ 4.0
```

A continuación se presentará un programa pequeño (rv1.tcl) que prueba una RV con distribución de Pareto con diferentes semillas y generadores pero con la misma distribución.

Para cada semilla (de valores 0, 1 y 2) y generador, se crea una secuencia de tres RVs. La variable "count" asigna el número de RVs que se crea usando "test" para cada semilla y generador.

Cuando se ejecuta este ejemplo se puede observar que para la semilla 0, los dos generadores dan valores diferentes de variable; así, se obtienen 6 valores diferentes (tres de cada generador). Para otras semillas, un generador crea tres valores diferentes, pero éstos no dependen del generador: el valor n -ésimo creado por el generador 1 es igual que el n -ésimo creado por el generador 2.

```
# Ejemplo de demostración del uso de RVs

set count 3

for {set i 0} {$i<3} {incr i} {

puts "==== i = $i "

set MyRng1 [new RNG]

$MyRng1 seed $i

set MyRng2 [new RNG]

$MyRng2 seed $i

set r1 [new RandomVariable/Pareto]

$r1 use-rng $MyRng1
$r1 set avg_ 10.0
$r1 set shape_ 1.2
puts stdout "Testing Pareto Distribution, avg = [$r1 set avg_]
shape = [$r1 set shape_]"
$r1 test $count

set r2 [new RandomVariable/Pareto]

$r2 use-rng $MyRng2
$r2 set avg_ 10.0
$r2 set shape_ 1.2
puts stdout "Testing Pareto Distribution, avg = [$r2 set avg_]
shape = [$r2 set shape_]"
$r2 test $count
}
```

Ejemplo 3.10 RVs con distribución de Pareto con diferentes semillas.

El resultado de ejecutar el ejemplo 3.10 es el que se muestra a continuación:

```
==== i = 0
Testing Pareto Distribution, avg = 10 shape = 1.2
3.657613e+00
3.407881e+00
1.829254e+01
Testing Pareto Distribution, avg = 10 shape = 1.2
1.828216e+00
1.713197e+00
2.184647e+00
==== i = 1
Testing Pareto Distribution, avg = 10 shape = 1.2
3.000649e+04
9.035939e+00
2.105085e+00
Testing Pareto Distribution, avg = 10 shape = 1.2
3.000649e+04
9.035939e+00
2.105085e+00
==== i = 2
Testing Pareto Distribution, avg = 10 shape = 1.2
1.684057e+04
5.071249e+00
2.915293e+00
Testing Pareto Distribution, avg = 10 shape = 1.2
1.684057e+04
5.071249e+00
2.915293e+00
```

3.4 EXTENSIONES A NS [1]

Antes de revisar cómo agregar funcionalidades extras a NS, es necesario examinar qué información está guardada y el directorio o archivo correspondiente. La figura 3.15 muestra parcialmente la estructura del directorio del simulador al instalar el paquete NS (Anexo A).

Entre los directorios `ns-allinone-2.1b` y `ns-2`, se encuentran las implementaciones del simulador (en C++ u OTcl), *scripts* OTcl para pruebas de validación y ejemplos de *scripts* OTcl. Dentro de este directorio, bajo otro llamado `tcl`, se encuentran todos los códigos OTcl, *scripts* de ejemplo y la mayoría de lo codificado en C++, las cuales implementan el planificador de eventos y las clases de los objetos componentes de red básicos, excepto las relacionadas con WWW, las cuales se encuentran en el nivel principal.

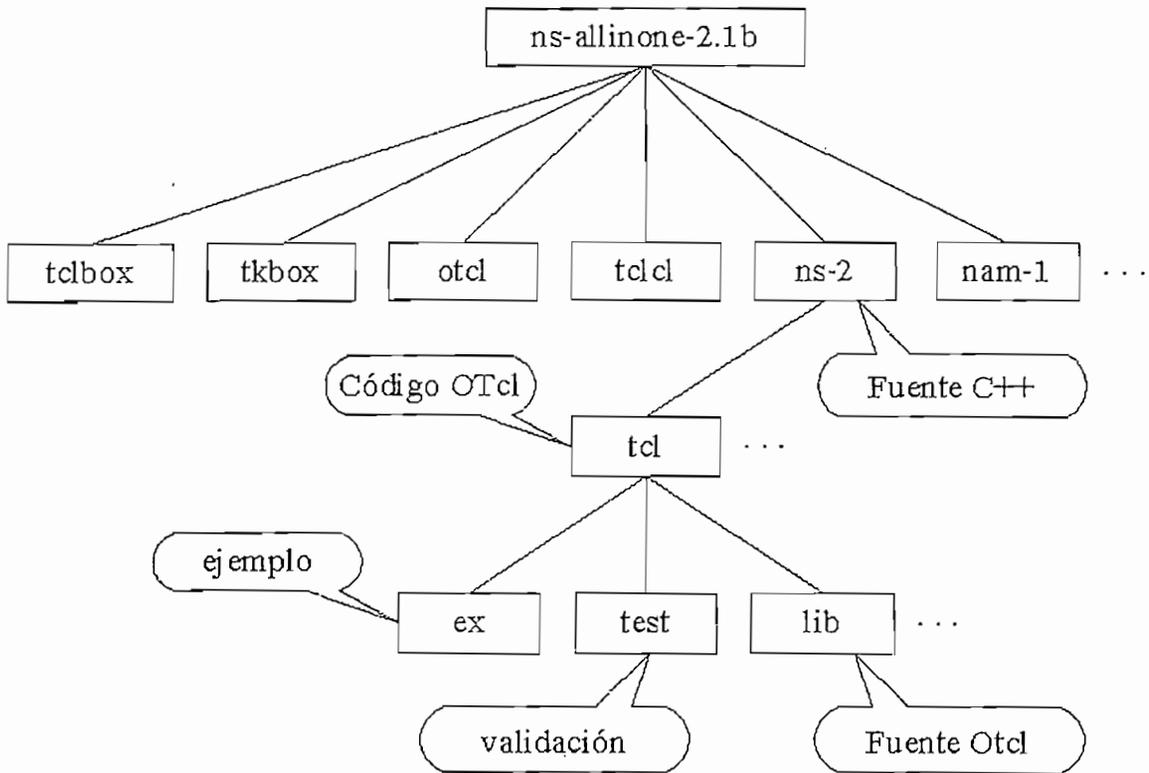


Figura 3.15 Estructura del directorio NS.

Por ejemplo, si se desea revisar la implementación del agente UDP, se debería ir al directorio “ns-allinone-2.1b/ns-2”, y abrir los archivos “udp.h”, “udp.cc” y todos los archivos que contienen las implementaciones de las clases ancestras necesitadas por UDP.

El directorio tcl tiene subdirectorios entre los cuales se encuentra lib que contiene los códigos fuente OTcl para muchas de las partes básicas y esenciales de la implementación NS (agentes, enlaces, nodos, paquetes, direcciones, ruteo, etc.), éste sería el lugar más visitado por usuarios o por desarrolladores.

Los códigos fuente OTcl de las implementaciones para LAN, Web y *Multicasting* se encuentran en subdirectorios separados de tcl. La siguiente es una lista parcial de archivos en el directorio “ns-2/tcl/lib” [2]:

- **ns-lib.tcl**: Aquí se encuentran la clase simulador y muchas de sus funciones miembro excepto para las relacionadas con LAN, Web y *Multicasting*.
- **ns-default.tcl**: Aquí se encuentran los valores por defecto de parámetros configurables para diversos componentes de red. Desde que muchos componentes de red son implementados en C++, los parámetros configurables son en realidad variables C++ habilitadas vía OTcl a través de la función de enlace "bind" (*nombre_de_variable_C++, nombre_de_variable_OTcl*).
- **ns-packet.tcl**: Aquí se encuentra la implementación de la inicialización del formato de cabecera del paquete. Cuando se crea una nueva cabecera de paquete, ésta se debería registrar en este archivo.
- **Otros archivos tcl**: Contienen implementaciones de objetos compuestos de red o de objetos de red inicio fin (parte de control) en C++. La aplicación FTP está completamente implementada en OTcl y el código fuente se encuentra en "ns-source.tcl".

3.4.1 ENCADENAMIENTO OTCL

Extender NS adhiriendo objetos de red básicos nuevos, generalmente involucra trabajar alrededor del encadenamiento entre OTcl y el código de C++, ya que la clase del objeto que se crea debe ser escrita en C++ por razones de eficiencia.

Esta sección presenta el encadenamiento disponible en NS a través de un ejemplo de creación de un simple y sencillo agente llamado "MyAgent" que no tiene un comportamiento de agente (no crea ni transmite paquetes).

Los ejemplos 3.11 a 3.14 muestran partes del código fuente C++ de "MyAgent" que juntos (con 3 líneas adicionales) hacen la implementación completa del agente.

3.4.1.1 Exportar una clase C++ a OTcl

Suponga que se desea crear una nueva clase de objeto de red en C++, dígase "MyAgent" que se deriva de la clase "Agent" y que se desea crear una instancia de este objeto en OTcl.

```
class MyAgent : public Agent {
public:
    MyAgent();
protected:
    int command(int argc, const char*const* argv);
private:
    int    my_var1;
    double my_var2;
    void   MyPrivFunc(void);
};
static class MyAgentClass : public TclClass {
public:
    MyAgentClass() : TclClass("Agent/MyAgentOtcl") {}
    TclObject* create(int, const char*const*) {
        return(new MyAgent());
    }
} class _my_agent;
```

Ejemplo 3.11 Definición de la clase "MyAgent" y encadenamiento con OTcl.

Para hacer esto, se debe definir un objeto de enlace, dígase "MyAgentClass", el cual debería estar derivado de "TclClass". Este objeto de enlace crea un objeto OTcl de nombre específico ("Agent/MyAgentOtcl" para el ejemplo), y crea un enlace entre el objeto OTcl y el objeto C++ ("MyAgent" en este ejemplo), del cual el procedimiento de "lanzamiento" está especificado en la función miembro **create**. El ejemplo 3.11 muestra la definición de la clase "MyAgent" y el enlace de ésta hacia OTcl.

Cuando se inicia NS por primera vez, se ejecuta el constructor para la variable estática "*class_my_agent*" y por lo tanto se crea una instancia de "MyAgentClass". En este proceso, la clase "Agent/MyAgentOtcl" y sus métodos apropiados (funciones miembro) se crean en el espacio OTcl. Siempre que un usuario en el espacio OTcl trate de crear una instancia de este objeto usando el

comando "new Agent/MyAgentOtcI", éste invocará a "MyAgentClass::create" que crea una instancia de "MyAgent" y devuelve la dirección. Note que crear un objeto de C++ desde OTcl no significa que se pueda invocar funciones miembro o variables miembro de acceso del objeto C++ desde OTcl.

3.4.1.2 Exportar variables de una clase C++ a OTcl

Supóngase un nuevo objeto C++, "MyAgent", que tiene dos parámetros variables, dígase "my_var1" y "my_var2", que se desean configurar (o cambiar) fácilmente desde OTcl usando un *script* de entrada de simulación. Para hacer esto se debe usar una función de lazo, o enlace, para cada una de las variables de la clase C++ que se quiera exportar.

Una función de enlace crea una nueva variable miembro del nombre dado (primer argumento) en el correspondiente objeto de la clase ("Agent/MyAgentOtcI"), y crea lazos bidireccionales entre la variable de la clase OTcl y la variable C++ cuya dirección está especificada como la segunda variable. El ejemplo 3.12 ilustra cómo crear lazos para "my_var1" y "my_var2" mostradas en el ejemplo 3.11.

```
MyAgent::MyAgent() : Agent(PT_UDP) {  
    bind("my_var1_otcl", &my_var1);  
    bind("my_var2_otcl", &my_var2);  
}
```

Ejemplo 3.12 Ejemplo de creación de una variable encadenada.

Note que las funciones enlazadas están localizadas en la función constructora "MyAgent" para establecer los lazos cuando una instancia de este objeto es creada. NS soporta cuatro tipos diferentes de funciones de enlace para cinco diferentes tipos de variables:

- `bind()`: variables reales o enteras
- `bind_time()`: variables de tiempo
- `bind_bw()`: variables de ancho de banda
- `bind_bool()`: variables booleanas

De esta manera, un usuario diseñará y ejecutará una simulación usando un *script* OTcl que pueda cambiar o acceder a parámetros configurables (o valores variables) de los componentes de red implementados en C++. Note que siempre que se exporta una variable C++, se recomienda que también se establezca el valor por defecto en el archivo "ns-2/tcl/lib/ns-lib.tcl"; de otra manera se encontrará un mensaje de precaución cuando se cree una instancia de un nuevo objeto.

3.4.1.3 Exportar comando de control de objeto de C++ a OTcl

Adicionalmente para exportar algunas de las variables objeto de C++, se desearía dar el control del objeto C++ a OTcl. Esto se logra definiendo una función miembro "command" del objeto C++ (MyAgent), la cual trabaja como un intérprete de comandos OTcl.

El ejemplo 3.13 muestra un ejemplo de la definición de la función miembro "command" del objeto "MyAgent" del ejemplo 3.11.

```
int MyAgent::command(int argc, const char*const* argv) {
    if(argc == 2) {
        if(strcmp(argv[1], "call-my-priv-func") == 0) {
            MyPrivFunc();
            return(TCL_OK);
        }
    }
    return(Agent::command(argc, argv));
}
```

Ejemplo 3.13 Ejemplo de intérprete de comandos OTcl.

Cuando se crea una instancia que corresponde al objeto "MyAgent" en el espacio OTcl (por ejemplo "set myagent [new Agent/MyAgentOtcl]"), y el usuario intenta llamar una función miembro de dicho objeto (por ejemplo "\$myagent call-my-priv-func"), OTcl busca el nombre de la función miembro dada en el objeto OTcl. Si el nombre de la función miembro dada no se encuentra, entonces invoca "MyAgent::command" pasando el nombre y los argumentos de la función miembro OTcl invocada en formato argc/argv.

Si existe una acción definida para el nombre de la función miembro invocada OTcl en la función miembro "command", ésta lleva la solicitud y devuelve un resultado. Si no, la función "command" para su objeto ancestro se llama recursivamente hasta que el nombre sea encontrado.

Si el nombre no se encuentra en ninguno de los ancestros, se devuelve un mensaje de error al objeto OTcl, y éste da uno similar al usuario. De esta manera un usuario en el espacio OTcl puede controlar el comportamiento de un objeto C++.

3.4.1.4 Ejecución de un comando OTcl desde C++

Así como se implementó un nuevo objeto de red en C++, se podría ejecutar un comando OTcl desde un objeto C++. El ejemplo 3.14 muestra la implementación de la función miembro "MyPrivFunc" de "MyAgent" en el ejemplo 3.11 el cual hace un intérprete OTcl imprimiendo los valores en las variables miembro privadas "my_var_1" y "my_var_2".

```
void MyAgent::MyPrivFunc(void) {  
  
    Tcl& tcl = Tcl::instance();  
    tcl.eval("puts \"Message From MyPrivFunc\"");  
    tcl.evalf("puts \"      my_var1 = %d\"", my_var1);  
    tcl.evalf("puts \"      my_var2 = %f\"", my_var2);  
  
}
```

Ejemplo 3.14 Ejecución de un comando OTcl desde un objeto C++.

Para ejecutar un comando OTcl desde C++, se debería conseguir una referencia a "Tcl::instance()" que es declarada como una variable miembro estática. Esto proporciona un par de funciones con las cuales se puede pasar un comando OTcl al intérprete (la primera línea de "MyPrivFunc" lo hace). Este ejemplo ilustra dos maneras de pasar un comando OTcl al intérprete.

3.5 NETWORK ANIMATOR (NAM) [4]

NAM es una herramienta basada en Tcl/Tk para observar los trazos de una simulación de red y los trazos de paquetes de una red real. Soporta capas de topologías, animación de paquetes por niveles y varias herramientas de inspección de datos.

El ejemplo 3.15 es un *script* de una sencilla simulación que almacena un archivo de trazo NAM, el cual se visualiza con la instrucción "nam out.nam".

```
set ns [new Simulator]

# Definición de colores para los diferentes flujos de datos
# (para el NAM)
$ns color 1 Blue
$ns color 2 Red

#Abre el archivo Trace
set tracefile1 [open out.tr w]
set winfile [open WinFile w]
$ns trace-all $tracefile1

#Abre el archivo NAM
set namfile [open out.nam w]
$ns namtrace-all $namfile

#Define el procedimiento 'finish'
proc finish {} {
    global ns tracefile1 namfile
    $ns flush-trace
    close $tracefile1
    close $namfile
    exec nam out.nam &
    exit 0
}

#Crea seis nodos
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

#Crea enlaces entre los nodos
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
```

```
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mb 30ms DropTail

#Posiciona los nodos en el NAM
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $ns orient right-down

#Establece el tamaño de cola del enlace (n2-n3) en 20
$ns queue-limit $n2 $n3 20

#Configuración de la conexión TCP
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set packetSize_ 552

#Configuración de FTP sobre una conexión TCP
set ftp [new Application/FTP]
$ftp attach-agent $tcp

#Configuración de una conexión UDP
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2

#Configuración de CBR sobre una conexión UDP
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packetSize_ 1000
$cbr set rate_ 0.01Mb
$cbr set random_ false

$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"

# Procedimiento para el ploteo del tamaño de ventana. Como
# argumentos necesita el nombre del nodo tcp fuente (llamado
# "tcpSource") y el archive de salida.
proc plotWindow {tcpSource file} {
global ns
set time 0.1
```

```
set now [$ns now]
set cwnd [$tcp Source set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 0.1 "plotWindow $tcp $winfile"

$ns at 125.0 "finish"
$ns run
```

Ejemplo 3.15 Script OTcl [3]

Cuando se ejecute el ejemplo 3.15, la herramienta de visualización NAM desplegará una red con 6 nodos. La localización de los nodos podría ser aleatoria. Para reproducir una localización inicial de los nodos como la de la figura 3.16, se agrega lo siguiente al programa tcl:

```
#Posicionamiento de los nodos en el NAM
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $ns orient right-down
```

Si una localización aleatoria de nodos es escogida y no es satisfactoria, se puede presionar el botón "re-layout" y escoger otra localización aleatoria. También se puede editar la localización haciendo clic en el botón Edit/View, y después en "draggind" en cada nodo para ubicarlo en el lugar deseado (con la ayuda del ratón).

En la figura 3.16 se puede observar que el NAM muestra la animación de los paquetes CBR (flujo del nodo 1 al 5) en rojo, y los paquetes TCP (fluyendo del nodo 0 a 4) en azul. Los TCP ACKs que van en direcciones inversas también están en azul pero de tamaño más pequeño, considerando que los ACK tienen un tamaño de 40 bytes en comparación con los paquetes TCP que tienen un tamaño de 552 bytes. Para obtener los colores, se tiene que definir al principio del ejemplo 3.15 lo siguiente:

```
$ns color 1 Blue
$ns color 2 Red
```

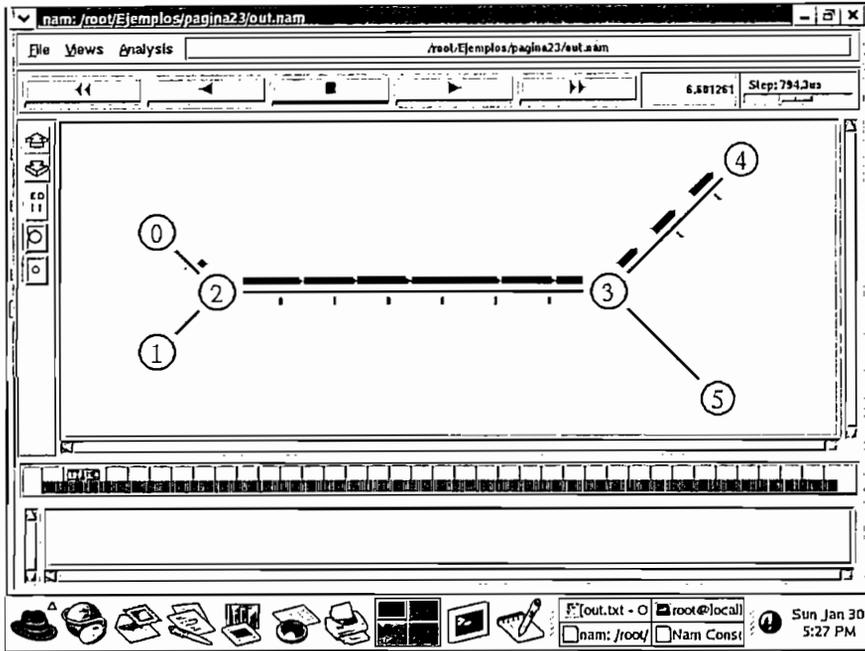


Figura 3.16 Visualización en el NAM del ejemplo 3.15 a los 6.6 segundos.

Otras cosas que pueden hacerse en NAM: [3]

- Color de los nodos; por ejemplo si se quiere que el n0 aparezca en rojo, se escribirá:

```
$ns n0 color red.
```

- Forma de los nodos; por defecto, los nodos son redondos, pero pueden aparecer con una forma diferente. Por ejemplo tecleando:

```
$n1 shape box (o en lugar de "box" se puede usar "hexagon" o "circle").
```

- Color de los enlaces; se tecléa por ejemplo:

```
$ns duplex-link-op $n0 $n2 color "green"
```

- Adherir y quitar marcas; se puede marcar un nodo en un momento dado (por ejemplo al mismo tiempo que se active alguna fuente de tráfico). Por ejemplo, se puede teclear:

```
$ns at 2.0 "$n3 add-mark m3 blue box"  
$ns at 30.0 "$n3 delete-mark m3"
```

Esto produce una marca azul que rodea el nodo 3 durante el intervalo de tiempo [2 s, 30 s]

- Adherir etiquetas; una etiqueta puede aparecer en pantalla en un tiempo dado, por ejemplo la etiqueta "active node" en el nodo n3 en el instante 1.2 s es habilitada tecleando:

```
$ns at 1.2 "$n3 lab \"active node\""
```

y para dar la etiqueta "TCP input link" al enlace n0-n2, se pone:

```
$ns duplex-link-op $n0 $n2 label "TCP input link"
```

- Adherir texto; al marco inferior de la ventana del NAM se puede hacer aparece en un momento dado algún texto. Esto se puede usar para describir algún evento que se fija en ese momento. Un ejemplo es:

```
$ns at 5 "$ns trace-annotate \"packet drop\""
```

- Se puede agregar además en el NAM un supervisor del tamaño de cola. Por ejemplo, supervisar la cola de entrada del enlace n2-n3 con:

```
$ns simplex-link-op $n2 $n3 queuePos 0.5
```

Referencias Bibliográficas del Capítulo 3

[1] <http://nile.wpi.edu/NS/>

[2] VINT Project, "*The ns Manual (formerly ns Notes and Documentation*", December 2003.

[3] Altman, E. y Jiménez, T., "*NS Simulator for beginners*", Lecture notes, 2003-2004, Univ. de Los Andes, Mérida, Venezuela and ESSI, Sophia-Antipolis, France. December 2003.

[4] <http://www.isi.edu/nsnam/nam/>