

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



“UN PROTOCOLO MAC (CONTROL DE ACCESO AL MEDIO)
EFICIENTE PARA REDES INALÁMBRICAS”

TESIS

PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS
MENCIÓN: TELECOMUNICACIONES

PRESENTADA POR
OMAR ERNESTO AZNARÁN TORRES

LIMA – PERÚ

2011

Con todo mi corazón dedico esta tesis a:

Mis padres Maura y Fernando, a mis
hermanas Rocío y Roxana, a mis
hermanos Fernando y Robert;

Nuestro Creador;

Milena.

El autor

Un agradecimiento muy especial al asesor de mi tesis MSc. Arturo Vilca Román, por sus consejos y el apoyo brindado, los cuales fueron muy útiles para llevar a cabo la programación del protocolo. Igualmente agradezco a los revisores de la tesis: MSc. Fernando Merchán y Dr. Arturo Rojas, quienes han colaborado con sus comentarios a darle claridad a este trabajo.

El autor

ABSTRACT

Packetized multiple accesses MAC protocols are the fundamental pieces for reaching those services that should be offered by IMT-2000 cellular networks [4]. After the initial proposal of PRMA protocol [1], many wireless MAC protocols were proposed to solve the collision problem in the uplink channel or to offer optimal techniques for resource scheduling. After studying the proposals mentioned above, we deduced that CRDA [2] offered the best performance about serving capacity (up to 144 mobile terminals) and also a good technique for solving collisions. However CRDA does not offer quality of service (QoS) support because manage all traffic data flows in a fair way. Besides [2] uses one single statistical model to represent data traffic and no differentiation is done about the traffic application type.

In this thesis work, we propose an efficient wireless MAC protocol with QoS support and with capacity of supporting data traffic from different classes (ITU M1079 [19]). For accomplishing our goal, the work is divided in three parts: First, we emulated the CRDA MAC protocol, [2], using OMNET++ discrete event simulator [3], after it was done we validated our protocol design. Second, we used four statistical distributions for simulating application traffic such as: conversational, interactive, unidirectional and underground classes. These new traffic models were used to prove the performance of CRDA. We checked that CRDA does not offer QoS and its serving capacity should be reduced to less than 120 terminals. Third, we implemented a QoS policy inside the scheduling algorithm of the base station module of our CRDA protocol programmed using OMNET++. The new MAC protocol (called ECRDA – enhanced CRDA) was tested using the new traffic models. The obtained results showed that ECRDA performs efficiently in comparison with CRDA by two reasons: first, it prioritizes critical data traffic avoiding that access delay overcomes the maximum threshold. Second, the serving capacity of ECRDA reaches up to 140 terminals, which is higher than what was reported with CRDA.

RESUMEN

Los protocolos MAC de acceso múltiple por paquetes para redes inalámbricas, son las piezas fundamentales, para alcanzar los servicios que deben ofrecer las redes celulares IMT-2000 [4]. Desde la aparición del protocolo MAC PRMA [1], surgieron una gran variedad de propuestas orientadas a resolver los problemas de colisión en el canal de subida y en ofrecer técnicas óptimas para la planificación y asignación de canales. Del estudio de dichas propuestas se deduce que CRDA [2] ofrece mejor desempeño en cuanto a la capacidad de atención (144 terminales móviles) y a la técnica usada para resolver las colisiones; sin embargo no ofrece soporte de calidad de servicio (QoS), ya que trata, a todos los flujos de tráfico de datos, de una manera equitativa. Además en [2] se usa un solo modelo estadístico para simular el tráfico de datos sin hacer diferenciación en el tipo de aplicación.

En este trabajo de tesis proponemos un protocolo MAC eficiente para redes inalámbricas, con soporte de QoS y con capacidad de atender tráfico de datos de distintas clases (UIT M1079 [19]). Para lograr nuestro objetivo se dividió el trabajo en tres partes: Primero, se ha implementado el protocolo de [2] usando el simulador de eventos discretos, OMNET++ [3], obteniéndose la misma tendencia estadística y validando el diseño propuesto. Segundo, se usaron cuatro distribuciones estadísticas para emular el tráfico de aplicaciones con clase: conversacional, interactiva, unidireccional y de fondo. Los nuevos modelos de tráfico sirvieron para probar el desempeño del protocolo CRDA, comprobándose que no ofrece QoS y que su capacidad de atención se ve reducida a menos de 120 terminales. Tercero, al protocolo CRDA programado en OMNET++, se le implementó una política de QoS dentro del algoritmo de scheduling, en la estación base. Luego se probó el nuevo protocolo MAC (llamado ECRDA) con los nuevos modelos de tráfico. Los resultados obtenidos demuestran que ECRDA se desempeña eficientemente en comparación a CRDA por dos razones: Primero, atiende prioritariamente los tráficos más sensibles y evita que el retardo de acceso supere el umbral permitido para cada aplicación. Segundo, la capacidad de atención de ECRDA llega hasta 140 terminales, superior a lo ofrecido por CRDA.

TABLA DE CONTENIDO

INDICE DE FIGURAS	VIII
INDICE DE TABLAS	XI
GLOSARIO DE ABREVIACIONES	XIII

CAPITULO 1 : GENERALIDADES

1.1	Introducción.....	1
1.2	Problemática Existente.....	2
1.3	Formulación del Problema	3
1.4	Solución Planteada	4
1.5	Antecedentes	4
1.6	Alcance del Trabajo.....	5
1.7	Contenido de la Tesis.....	6

CAPITULO 2 : DISEÑO E IMPLEMENTACIÓN DEL PROTOCOLO MAC

2.1	Marco Teórico	8
2.1.1	Protocolos MAC para redes inalámbricas celulares	8
2.1.2	Modelos estadísticos para las fuentes de tráfico	9
2.1.2.1	Modelo para Voz	9
2.1.2.2	Modelo para Datos	11
2.1.3	Clases de QoS en las IMT-2000.....	13
2.1.4	Simulador de Eventos Discretos OMNET++.....	15
2.1.4.1	Los archivos requeridos por OMNET++	15
2.2	Diseño del protocolo MAC.....	18
2.2.1	Estructura de la Trama	18
2.2.1.1	Parámetros de Diseño.....	19
2.2.1.2	Fórmulas de Diseño	24
2.2.1.3	Estructura Interna de las Ranuras	25
2.2.2	Fuentes de Tráfico Utilizadas	27
2.2.3	Funcionamiento del Protocolo	32
2.2.3.1	Funcionamiento para Voz.....	35

2.2.3.1.1 Fase de Reserva de Canal	35
2.2.3.1.2 Fase de Planificación	37
2.2.3.2 Funcionamiento para Datos de Velocidad Binaria Variable.....	39
2.2.3.2.1 Fase de Reserva de Canal	40
2.2.3.2.2 Fase de Planificación	43
2.2.4 Diagramas de Señalización, y Temporizadores.....	48
2.2.5 Métricas usadas.....	51
2.3 Implementación del protocolo MAC en OMNET++	53
2.3.1 Ambiente de Simulación	54
2.3.2 Módulos del Protocolo	58
2.3.3 Parámetros de la Simulación	76
2.3.4 Restricciones del Trabajo de Tesis.....	79

CAPITULO 3 : PRUEBAS REALIZADAS Y RESULTADOS OBTENIDOS

3.1 Requerimientos computacionales en la simulación.....	81
3.2 Pruebas realizadas y Resultados obtenidos	81
3.2.1 Primer grupo de pruebas.....	81
3.2.2 Segundo grupo de pruebas	89
3.2.3 Tercer grupo de pruebas.....	91

CAPITULO 4 : CONCLUSIONES

4.1 CONCLUSIONES	96
4.2 FUTUROS TRABAJOS DE INVESTIGACIÓN	97

BIBLIOGRAFÍA	99
--------------------	----

ANEXOS

ANEXO A : PROTOCOLOS MAC PARA REDES INALÁMBRICAS	102
ANEXO B : MODELAMIENTO DE FUENTES DE TRÁFICO PARA DATOS	110
ANEXO C : CALIDAD DE SERVICIO (QOS)	118
ANEXO D : DETALLES DE OMNET++.....	120
ANEXO E : SOFTWARE DE LA TESIS	125

INDICE DE FIGURAS

CAPITULO 2

FIGURA 2.1	MODELO PARA VOZ, UNA CADENA DE MARKOV DE DOS ESTADOS: ACTIVIDAD (ON) Y SILENCIO (OFF), FUENTE [11]	10
FIGURA 2.2	MODELO ON/OFF, FUENTE [16]	12
FIGURA 2.3	ENLACE DE ARCHIVOS DENTRO DE OMNET++, FUENTE [3], OMNET++ MANUAL AÑO 2005	17
FIGURA 2.4	ESTRUCTURA DE LA TRAMA TDMA DEL PROTOCOLO MAC EFICIENTE, FUENTE [2]	19
FIGURA 2.5	POSICIONES DE LOS R-, A-, I-SLOTS PARA DOS TRAMAS DE SUBIDA Y BAJADA SINCRONIZADAS, FUENTE [2]	21
FIGURA 2.6	EFECTO DE FRONTERA EN LA ASIGNACIÓN DE I-SLOT-UP, FUENTE [2]	23
FIGURA 2.7	ESTRUCTURA INTERNA DE LAS RANURAS	26
FIGURA 2.8	MODELO PARA UNA FUENTE DE TRÁFICO INTERACTIVA, FUENTE [20]	29
FIGURA 2.9	MODELO DE TRÁFICO PARA FLUJO DE VIDEO, FUENTE [20]	31
FIGURA 2.10	ESQUEMA CON LOS ALGORITMOS RESIDENTES EN CADA OBJETO (TERMINAL MÓVIL Y ESTACIÓN BASE)	33
FIGURA 2.11	EVOLUCIÓN DEL ESTADO EN EL TERMINAL MÓVIL	34
FIGURA 2.12	FASE DE RESERVA DE CANAL – LADO DEL TERMINAL MÓVIL (FUNCIONAMIENTO PARA VOZ, PARTE A)	36
FIGURA 2.13	FASE DE RESERVA DE CANAL – LADO DEL TERMINAL MÓVIL (FUNCIONAMIENTO PARA VOZ, PARTE B)	36
FIGURA 2.14	FASE DE RESERVA DE CANAL – LADO DEL TERMINAL MÓVIL (FUNCIONAMIENTO PARA VOZ, PARTE C)	37
FIGURA 2.15	FASE DE PLANIFICACIÓN – LADO DE LA ESTACIÓN BASE (FUNCIONAMIENTO PARA VOZ)	38
FIGURA 2.16	FASE DE RESERVA DE CANAL – LADO DEL TERMINAL MÓVIL (FUNCIONAMIENTO PARA DATOS, PARTE A)	40
FIGURA 2.17	FASE DE RESERVA DE CANAL – LADO DEL TERMINAL MÓVIL (FUNCIONAMIENTO PARA DATOS, PARTE B)	42
FIGURA 2.18	FASE DE PLANIFICACIÓN – LADO DE LA ESTACIÓN BASE (FUNCIONAMIENTO PARA DATOS – PARTE A)	44
FIGURA 2.19	FASE DE PLANIFICACIÓN – LADO DE LA ESTACIÓN BASE: ESTRATEGIA QOS, CLASIFICACIÓN DE SIGS	45
FIGURA 2.20	FASE DE PLANIFICACIÓN – LADO DE LA ESTACIÓN BASE: ESTRATEGIA DE PIGGYBACKING	47

FIGURA 2.21	FASE DE PROGRAMACIÓN – LADO DE LA ESTACIÓN BASE: (FUNCIONAMIENTO PARA DATOS – PARTE B).....	48
FIGURA 2.22	SEÑALIZACIÓN ENTRE MS Y BS – CASO VOZ.....	49
FIGURA 2.23	SEÑALIZACIÓN ENTRE MS Y BS – CASO DATOS DE VELOCIDAD BINARIA VARIABLE	50
FIGURA 2.24	TEMPORIZADOR – PRIMER CASO	52
FIGURA 2.25	TEMPORIZADOR – SEGUNDO CASO.....	52
FIGURA 2.26	TOPOLOGÍA EN OMNET++.....	54
FIGURA 2.27	INTEGRACIÓN DEL PROTOCOLO CON EL FRAMEWORK DE OMNET++.....	57
FIGURA 2.28	MÓDULOS DEL PROTOCOLO – MÓDULOS SIMPLES Y MÓDULOS COMPUESTOS	59
FIGURA 2.29	MÓDULOS ESTACIÓN BASE (IZQ.) Y TERMINAL MÓVIL (DER.) EN OMNET++.....	59
FIGURA 2.30	MENSAJES EN OMNET++	63
 CAPITULO 3		
FIGURA 3.1	CDF DEL RETARDO DE ACCESO, COMPARA EL CDF DEL PAPER, [2] VS DIFERENTES VALORES DE λ , PARA UNA SIMULACIÓN DE 140 TERMINALES: 100V Y 40D.....	86
FIGURA 3.2	PROMEDIO POR RETARDO DE ACCESO PARA TERMINALES DE DATOS - SIMULACIÓN VS [2]	89
 ANEXO A		
FIGURA A.1	PRMA, FUENTE [1]	104
FIGURA A.2	PRMA++, FUENTE [9]	104
FIGURA A.3	SIR, FUENTE [10].....	105
FIGURA A.4	CRDA, FUENTE [2]	107
 ANEXO B		
FIGURA B.1	TRAFICO AUTOSIMILAR, FUENTE [14].....	113
FIGURA B.2	DISTRIBUCIONES DE COLA PESADA: PARETO AND EXPONENTIAL PROBABILITY DENSITY FUNCTIONS, FUENTE [14].....	115
FIGURA B.3	PROCESO DE DISEÑO DE UN MODELO, FUENTE [12]	116
FIGURA B.4	PROCESO DE SIMULACIÓN, FUENTE [12].....	117

ANEXO D

FIGURA D.1	JERARQUÍA DE LOS MÓDULOS EN OMNET++, FUENTE [3], OMNET++ MANUAL AÑO 2005	120
FIGURA D.2	MÓDULOS Y SUB MÓDULOS CONECTADOS MEDIANTE PUERTAS Y ENLACES, FUENTE [3], OMNET++ MANUAL AÑO 2005.....	121
FIGURA D.3	ARQUITECTURA DE LOS PROGRAMAS DE SIMULACIÓN EN OMNET++, FUENTE [3], OMNET++ MANUAL AÑO 2005.....	123

INDICE DE TABLAS

CAPITULO 2

TABLA 2.1	CLASES DE QOS PARA LAS IMT-2000, FUENTE [19].....	14
TABLA 2.2	GRUPOS DE COMPORTAMIENTO DE APLICACIONES SEGÚN LAS NECESIDADES DE QOS, FUENTE [19].....	14
TABLA 2.3	PARÁMETROS OPERACIONALES DEL PROTOCOLO MAC EFICIENTE.....	76
TABLA 2.4	PARÁMETROS PARA VOZ	77
TABLA 2.5	PARÁMETROS PARA DATOS SIN PRIORIDAD.....	77
TABLA 2.6	PARÁMETROS PARA CLASE CONVERSACIONAL.....	78
TABLA 2.7	PARÁMETROS PARA CLASE INTERACTIVA	78
TABLA 2.8	PARÁMETROS PARA CLASE UNIDIRECCIONAL	79

CAPITULO 3

TABLA 3.1	RESULTADOS DE LAS PRUEBAS CON LAS FUENTES DE TRÁFICO Y EL TIEMPO DE SIMULACION	83
TABLA 3.2	RESULTADOS DE LAS PRUEBAS PARA HALLAR UN VALOR OPTIMO DE LAMBDA.....	85
TABLA 3.3	PROMEDIO POR RETARDO DE ACCESO PARA TERMINALES DE DATOS	88
TABLA 3.4	CRDA / DESEMPEÑO CON FUENTES DE TRÁFICO QUE DEMANDAN DISTINTOS REQUERIMIENTOS DE QOS - PROMEDIO POR RETARDO DE ACCESO (PRA)	90
TABLA 3.5	CRDA / DESEMPEÑO CON FUENTES DE TRÁFICO QUE DEMANDAN DISTINTOS REQUERIMIENTOS DE QOS – VALOR PICO POR RETARDO DE ACCESO	91
TABLA 3.6	ECRDA VS CRDA / MÉTRICA DE COMPARACIÓN - PROMEDIO POR RETARDO DE ACCESO (PRA)	92
TABLA 3.7	ECRDA VS CRDA / MÉTRICA DE COMPARACIÓN – VALOR PICO POR RETARDO DE ACCESO	93
TABLA 3.8	ECRDA / MÉTRICA DE COMPARACIÓN PRA – CAPACIDAD PARA ATENDER CIERTA CANTIDAD DE TERMINALES MÓVILES	94
TABLA 3.9	ECRDA / MÉTRICA DE COMPARACIÓN VALOR PICO POR RETARDO DE ACCESO - CAPACIDAD PARA ATENDER CIERTA CANTIDAD DE TERMINALES MÓVILES	95

ANEXO C

TABLA C.1 DIMENSIONES PARA EL DESEMPEÑO DE LA QOS POR
 APLICACIÓN..... 119

ANEXO E

TABLA E.1 SIMULACIONES DE LA TESIS..... 126

GLOSARIO DE ABREVIACIONES

3G	: Third Generation Mobile Services (servicios móviles de tercera generación)
3G+	: Future Developments based on Third Generation Mobile Services (desarrollos futuros basados en los servicios móviles de tercera generación)
3GPP	: 3rd Generation Partnership Project (socios del proyecto de tercera generación)
4G	: Fourth Generation Mobile Services, also Beyond 3G (servicios móviles de cuarta generación, más allá de 3G)
ACK	: Acknowledgment (reconocimiento)
BS	: Base Station (estación base)
CDF	: Cumulative Distribution Function (función de distribución acumulativa)
CDMA	: Code Division Multiple Access (acceso múltiple por división de código)
CRDA	: Collision Resolution and Dynamic Allocation MAC Protocol (protocolo MAC con resolución de colisiones y asignación dinámica)
ECRDA	: Enhanced CRDA (protocolo CRDA mejorado; propuesto y simulado en esta tesis)
FCFS	: First Coming, First Served (el primero que ingresa es el primero que se atiende)
FDD	: Frequency Division Duplexing (doblado por división de frecuencia)
FDMA	: Frequency Division Multiple Access (acceso múltiple por división de frecuencia)
FIFO	: First Input First Output (primero en entrar, primero en salir)
FTP	: File Transfer Protocol (protocolo de transferencia de archivos)
GSM	: Global System for Mobile Communications (sistema global para comunicaciones móviles)

IMT-2000	: International Mobile Telecommunication - 2000 (telecomunicaciones móviles internacionales 2000)
IP	: Internet Protocol (protocolo internet)
MAC	: Medium Access Protocol (protocolo de acceso al medio)
MMPP	: Markov Modulated Poisson Process (proceso de Poisson modulado por una cadena de Markov)
MS	: Mobile Station (estación móvil)
NSC	: Network Service Classes (clases de servicio de red)
PRA	: Promedio por Retardo de Acceso
PSTN	: Red Telefónica Pública Conmutada
QoS	: Quality of Service (calidad de servicio)
RR	: Round Robin (cola de atención cíclica)
S-CDMA	: CDMA síncrono
TDD	: Time Division Duplexing (doblado por división de códigos)
TDMA	: Time Division Multiple Access (acceso múltiple por división de tiempo)
VAD	: Voice Activity Detection (detección de la actividad de voz)
VBR	: Variable Bit Rate (velocidad binaria variable)
WWW	: World Wide Web (internet)

CAPÍTULO 1

GENERALIDADES

1.1 Introducción

La recomendación M.1035 de la UIT, [4], para sistemas móviles de tercera generación (3G) define la capa de control de acceso al medio (MAC) como la encargada de controlar el enlace radioeléctrico de la capa física y realizar el control de calidad del enlace y la correspondencia del flujo de datos en el enlace radioeléctrico. Esta recomendación también sugiere que el diseño de un protocolo MAC debe tomar en cuenta los siguientes aspectos:

- La asignación del ancho de banda se debe realizar mediante un algoritmo de planificación (scheduling), el cual debe considerar como variables: la carga del sistema, los modelos de tráfico, los tipos de servicio, las prioridades del servicio y la situación de interferencia.
- La calidad de funcionamiento se debe lograr aplicando técnicas de control de velocidad para transmisión por paquetes de datos variable, además que dichas técnicas deben soportarse directamente sobre la interfaz aérea.
- El mejoramiento de la capacidad del canal se puede lograr utilizando técnicas de asignación dinámica de canal ó transmisión discontinua de la voz (VAD).

De otro lado, la recomendación que marca la pauta para futuros desarrollos de tecnologías 3G y aquellos sistemas más allá de 3G (Cuarta Generación – 4G), [18], sugiere el soporte de nuevas aplicaciones basados en IP, debido a la creciente demanda de los usuarios móviles en cursar tráfico de tipo multimedia, que tiene como característica un mayor consumo de ancho de banda y exigencias en la calidad del servicio (QoS). En cuanto a la QoS, la recomendación UIT M1079, [19], sugiere clasificar el tráfico de paquetes en cuatro clases: clase conversacional, clase interactiva, clase unidireccional y clase de fondo. Cada clase

mencionada exige un requerimiento distinto en cuanto al retardo de acceso, pérdida de paquetes y ancho de banda mínimo.

Así mismo se realizó el estudio y análisis de diversos protocolos MAC inalámbricos enfocados en 3G, encontrándose que CRDA [2] presenta el mejor desempeño (Apéndice A, sección A.2). Sin embargo [2] no ofrece soporte de calidad de servicio porque trata a todos los paquetes de datos de una manera equitativa.

Tomando en cuenta las recomendaciones [4], [17-19] y luego de haber estudiado algunos protocolos MAC para 3G, planteamos en esta tesis un protocolo MAC eficiente para redes inalámbricas, el cual ha sido programado en C++ usando el simulador de eventos discretos OMNET++, [3].

1.2 Problemática Existente

Dentro de una red inalámbrica celular como las redes de 3G, la comunicación entre la estación base y los terminales móviles se realiza a través de un canal de difusión, el cual se asigna de manera aleatoria. La problemática de la asignación del canal para la comunicación entre la estación base y los terminales móviles puede tratarse en dos partes: el problema en la asignación del canal de subida (desde el terminal móvil hacia la estación base) y el problema en la asignación del canal de bajada (desde la estación base hacia el terminal móvil). Con respecto al canal de bajada, el problema de acceso al canal es resuelto por la estación base quien tiene el control de la transmisión. Pero en el canal de subida surge el problema cuando más de un terminal móvil intenta transmitir al mismo instante usando el mismo canal de acceso aleatorio; generando de esta manera una colisión.

Los protocolos MAC actuales permiten que los terminales móviles envíen a la estación base una solicitud previa por uso de canal, para acceder a los canales de subida. Obviando por un instante el problema de la colisión, al asumir que un terminal móvil pudo enviar su solicitud por uso de canal de manera exitosa (sin

colisión), surgen otras problemáticas asociadas a la fase de planificación (scheduling) del protocolo MAC.

La fase de planificación se encarga de definir cuantos canales se asignarán al terminal móvil, que prioridad de atención se le dará, cuál será la velocidad mínima requerida por el terminal. Todas estas preguntas están en estrecha relación al tipo de tráfico de paquetes enviado por el terminal móvil.

Para completar la idea, el tráfico de paquetes que se cursa en una red de 3G proviene de aplicaciones con diferentes demandas de velocidad, retardo y pérdida de paquetes, volviéndose necesario contar con una política de calidad de servicio dentro del diseño del protocolo MAC inalámbrico.

1.3 Formulación del Problema

En el segmento de subida, el tráfico que desea enviar el terminal móvil hacia la estación base, demanda diversos requisitos en cuanto al retardo, velocidad y pérdida de paquetes. Esta situación origina un problema en la asignación de recursos ó canales de subida que obliga a idear un algoritmo de planificación (ó scheduling) que tome en cuenta el tipo de tráfico solicitante, la prioridad y la cantidad de canales que demanda. Estamos hablando de ofrecer calidad de servicio en el funcionamiento del protocolo MAC.

Los protocolos MAC estudiados (Apéndice A) proponen soluciones para resolver las disputas por el uso del canal en el segmento de subida y ofrecen soluciones para asignar los canales de una manera equitativa cuando se trata de tráfico de paquetes de datos. El problema es que dichos protocolos no abordan el tema de calidad de servicio y tampoco han sido probados con modelos de fuentes de tráfico más actuales como las sugeridas en la recomendación UIT M1079 [19].

En este trabajo de tesis nos centramos en el diseño, desarrollo y simulación de un protocolo MAC que sea eficiente al proveer soporte en calidad de servicio (QoS) para las clases de tráfico sugeridas en [19].

1.4 Solución Planteada

La solución que planteamos al problema formulado en la sección 1.3, consiste en programar un protocolo MAC eficiente para redes inalámbricas usando el simulador de eventos discretos OMNET++ [3] y tomando como base el protocolo MAC propuesto en [2], llamado CRDA. Luego siguiendo la recomendación UIT M1079 [19] planteamos el reemplazo del modelo para fuentes de tráfico de datos que se usó en [2], por nuevos modelos de tráfico como los propuestos en [20], [22] y que se enmarcan dentro de la recomendación [19]. La finalidad de esto último es lograr un escenario de simulación más cerca de la realidad y en donde las demandas por cantidad de canales de subida serán distintas. Los nuevos modelos de tráfico usan distribuciones estadísticas parametrizadas y emulan aplicaciones particulares como juegos en línea, video unidireccional y búsqueda web.

Para lograr la calidad de servicio, planteamos la inclusión de un esquema de priorización de tráfico, dentro del módulo que se encarga de asignar los canales de subida en la estación base. Con esto se espera clasificar y priorizar la atención de paquetes en el siguiente orden: primero se atiende la clase conversacional (voz, juegos en línea), segundo se atiende la clase interactiva (búsqueda web), tercero se atiende la clase unidireccional (video streaming) y por último se atiende la clase de fondo.

Finalmente sostenemos que nuestra propuesta de protocolo MAC para redes inalámbricas logrará soportar eficientemente las demandas en retardo y ancho de banda que imponen las nuevas fuentes de tráfico [19] usadas.

1.5 Antecedentes

Existen diferentes propuestas de protocolos MAC para redes inalámbricas cuya aplicación se orienta a las redes de paquetes de tercera generación y siendo estos los siguientes: PRMA, PRMA++, SIR, CRDA. Para la realización de la presente tesis se han estudiado los mencionados protocolos; en resumen, dichos protocolos abordan el tema de las colisiones ocasionadas cuando más de un terminal móvil intenta transmitir un paquete a la estación base; también abordan el tema de diferenciar el servicio entre voz y datos, además de ofrecer técnicas para asignar

los canales (time slots) por demanda. En el Anexo A, sección A.1, se presenta un listado de los protocolos MAC mencionados con una breve descripción de los mismos y en la sección A.2, se ha realizado una comparación de sus ventajas y desventajas. Del análisis realizado deducimos que el protocolo CRDA [2] ofrece mejor desempeño en cuanto a la capacidad de atención (144 terminales) y a la técnica usada para resolver las colisiones; es por ello que lo tomamos como protocolo MAC referencial. Cabe mencionar que en ninguno de los trabajos de la sección A.1, se ofrece calidad de servicio a las fuentes de tráfico y tampoco dichos protocolos se han probado con modelos de tráfico que demandan distintas velocidades de transmisión, distintos retardos máximos permitidos ó distintas tolerancias a la pérdida de paquetes. También resaltamos la ausencia del código fuente en los trabajos mencionados, lo cual ha obligado a que se desarrolle el protocolo desde cero.

1.6 Alcance del Trabajo

El presente trabajo de tesis se enfoca en el diseño, programación, simulación y evaluación de un protocolo MAC inalámbrico para redes celulares de tercera generación a más. Para este propósito se ha utilizado el simulador de eventos discretos OMNET++ [3], que permite modelar el comportamiento del protocolo diseñado, así como también modelar las fuentes de tráfico usando distribuciones estadísticas. Además, las fuentes de tráfico han sido clasificadas en función de los requerimientos que solicitan, tales como: ancho de banda (cantidad de ranuras de tiempo de subida) y retardo máximo permitido.

Técnicamente hablando, se simula una estación base con una cierta cantidad de terminales móviles que compiten por acceder a uno o varios spots (ranuras) de la trama de subida. La asignación de el (los) slot(s) se realiza en función del tipo de tráfico que lo solicita y de la prioridad que tiene dicho tráfico para ser atendido. Nuestro protocolo emula una trama TDMA de 72 ranuras de tiempo tanto en el segmento de subida como en el de bajada. El trabajo se enfoca en el segmento de subida. Para minimizar las colisiones se asume el uso de CDMA síncrono en determinados intervalos de tiempo dentro de los cuales solo se permite realizar

peticiones para reserva de slots de tiempo. En cuanto a la calidad de servicio, nuestro protocolo clasifica el tráfico de paquetes de datos y asigna los slots de tiempo usando cuatro clases diferenciadas: clase conversacional, clase interactiva, clase unidireccional y clase de fondo. El tráfico de paquetes de voz se atiende con la prioridad más alta y no compite y tampoco se clasifica dentro de las cuatro categorías mencionadas anteriormente. Para simular las diferentes clases de tráfico se han seleccionado modelos de distribuciones estadísticas de tres aplicaciones particulares: juegos en línea que clasifica dentro de la clase conversacional, búsqueda web que clasifica dentro de la clase interactiva y video streaming que clasifica dentro de la clase unidireccional. Para la clase de fondo se utilizó el modelo de tráfico de datos propuesto en [2].

El presente trabajo de tesis también permite medir el impacto que tienen las aplicaciones de datos en el desempeño del protocolo MAC y como podemos mejorar el funcionamiento al añadir una política de calidad de servicio (QoS).

1.7 Contenido de la Tesis

La presente tesis se estructura de la siguiente manera:

Dentro del Capítulo 2: La sección 2.1 aborda el marco teórico detallando los conceptos necesarios para entender el presente trabajo de tesis. En general se trata acerca de las características de los protocolos MAC para redes inalámbricas, las características de las fuentes de tráfico, la categorización de la calidad de servicio (QoS) en las IMT-2000 y detalles relevantes del simulador OMNET++. La sección 2.2 presenta el diseño del protocolo MAC eficiente de esta tesis. Dentro de esta sección se detallan el formato de las tramas TDMA, las fuentes de tráfico utilizadas, y el funcionamiento del protocolo soportado mediante modelos de diagramas de flujo y de señalización. La sección 2.3 detalla como se implementó el protocolo de esta tesis dentro del Framework de OMNET++. En el Capítulo 3 se detallan las pruebas realizadas y resultados obtenidos Finalmente, el Capítulo 4 se presentan las conclusiones de la tesis.

CAPÍTULO 2

DISEÑO E IMPLEMENTACIÓN DEL PROTOCOLO MAC

Debido a la demanda de los usuarios de telefonía móvil, por el uso de aplicaciones IP (como por ejemplo: juegos en línea, búsqueda web, video streaming) que generan tráfico de paquetes con distintos requerimientos de QoS, se hace imperioso contar con un protocolo de capa MAC que brinde el soporte a velocidades binarias variables y que además garantice la aplicación de una política de QoS. Es así que, el protocolo MAC propuesto en esta tesis, soporta el transporte de tráficos con diferentes demandas de QoS, como los que se han clasificado en [19]: tráfico conversacional, interactivo, unidireccional y de fondo. De esta manera el presente trabajo se enmarca dentro de los requerimientos solicitados por las normas UIT-R, recomendaciones M.1645 y M.1079-2, [18,19]

Para lograr nuestro objetivo, se tomó como referencia para el diseño el protocolo CRDA, (las razones se detallan en el Anexo A, sección A.2), y luego se realizaron dos modificaciones:

- La primera modificación consistió en reemplazar las fuentes de tráfico para datos, por modelos que se ajustan a la recomendación [19]
- La segunda modificación consistió en programar una política de priorización de tráfico dentro del algoritmo de asignación de canales, el cual reside en la estación base.

Para poder diseñar el protocolo MAC eficiente de esta tesis (llamado ECRDA) se estudió detalladamente el funcionamiento de CRDA y luego se infirió y levantó la información que ayudó a la implementación de ECRDA dentro del entorno de OMNET++ [3]. La información levantada e inferida consistió en:

- Los parámetros de diseño
- Las fórmulas de diseño

- Los diagramas de flujo del funcionamiento de CRDA
- Los diagramas de señalización y temporizadores del protocolo

La implementación de ECRDA en OMNET++ implicó materializar la información inferida en el diseño y portarla a la estructura del simulador; esto último significó programar el protocolo ECRDA usando lenguaje C++ e integrar el programa a las librerías del simulador.

A continuación se abordan en este capítulo los detalles del diseño e implementación del protocolo MAC eficiente ECRDA, no sin antes exponer algunos conceptos teóricos que serán muy útiles para la comprensión de este capítulo.

2.1 Marco Teórico

2.1.1 Protocolos MAC para redes inalámbricas celulares

De acuerdo a [5], los protocolos MAC para redes inalámbricas se caracterizan por:

- Los métodos para repartir el recurso (también llamado canal)
- Las técnicas de acceso múltiple
- La técnica de doblado utilizada para acceder al medio físico.

En particular, cuando se habla de redes celulares, los protocolos MAC expuestos en el Anexo A - sección A.1, se basan en el acceso por reserva de paquetes (PRMA) y asignan los recursos por demanda. Esta técnica es muy eficiente cuando se transmite tráfico con velocidades binarias variables porque administra mejor los recursos, evita el desperdicio de ancho de banda y disminuye la competencia por el uso del canal. Para información adicional sobre los métodos para repartir el recurso ó canal puede revisar la referencia [5].

En cuanto a la técnica de acceso múltiple, los protocolos del Anexo A – sección A.1, usan TDMA, FDMA, CDMA y en algunos casos, como en [2], combinan dichas técnicas a fin de explotar la capacidad del canal al máximo. Una alternativa

actual es el uso de OFDMA. Para información adicional sobre las técnicas de acceso múltiple puede revisar la referencia [6].

La técnica de doblado indica si tanto el terminal móvil (MS) como la estación base (BS), serán capaces de transmitir y recibir sobre el mismo canal. Los protocolos MAC del Anexo A – sección A.1 usan la técnica FDD, sin embargo en [7] se hace un estudio de protocolos MAC que además usan técnicas basadas en TDD.

2.1.2 Modelos estadísticos para las fuentes de tráfico

Debido a que las redes de 3G y aquellas más allá de 3G (en inglés beyond 3G) deben tener como requisito fundamental soportar servicios de velocidad variable ([4] – apartado 12.3); la mejor manera de representar dichos servicios, es a través de modelos estadísticos de tráfico que reflejen el comportamiento de las aplicaciones y de la manera más real posible. A continuación se explican los modelos matemáticos usados para representar el comportamiento de las fuentes de tráfico para voz y datos.

2.1.2.1 Modelo para Voz

De acuerdo al modelo de Brady [11] una fuente de voz puede modelarse mediante una cadena de Markov de dos estados, como se muestra en la figura 2.1.

Es así que se define a σ como la probabilidad de transición de un paso (en inglés one-step) que va desde el estadio de “silencio” hacia el estadio de “actividad”. De la misma forma, γ se define como la probabilidad de transición de un paso en la dirección opuesta; es decir desde el estadio de “actividad” hacia el estadio de “silencio”.

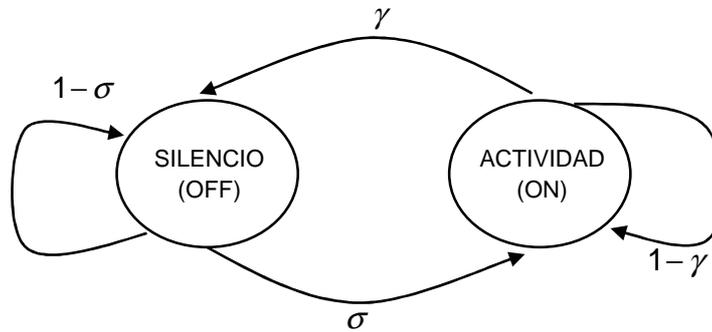


Figura 2.1 Modelo para Voz, una cadena de Markov de dos estados: actividad (ON) y silencio (OFF), Fuente [11]

De esta manera el modelo que se tiene se resume a:

$$\sigma = 1 - e^{-\left(\frac{\tau}{t_S}\right)} \quad (2.1)$$

$$\gamma = 1 - e^{-\left(\frac{\tau}{t_T}\right)} \quad (2.2)$$

Donde t_S y t_T representan respectivamente, el tiempo medio de permanencia de una fuente en estado de “silencio” y en estado de “actividad”, y τ es el intervalo entre dos instantes consecutivos.

En resumen, el modelo de Brady permite modelar una fuente de voz a la salida de un detector de actividad de voz (VAD). Esto último permite la transmisión de voz por paquetes, ahorrando ancho de banda y dejando de lado el modelo tradicional de circuito conmutado en donde el canal se desperdicia al transmitir períodos de silencio durante una conversación.

2.1.2.2 Modelo para Datos

Durante mucho tiempo se ha modelado el tráfico de datos mediante el modelo de Poisson, el cual se caracteriza por la ausencia de ráfagas (en inglés bursty) y la baja variabilidad, reflejada en la independencia de las muestras. El problema del modelo de Poisson es que no mantiene una correlación entre los tiempos de arribo de paquetes, siendo de esta manera, un proceso sin memoria que no depende de las muestras pasadas. Los estudios realizados en [13] demuestran que el tráfico de internet posee características de auto similitud, lo cual hace suponer la elección de modelos estadísticos de cola pesada y dependencia de largo plazo cada vez que se desea simular un generador de tráfico para datos. Información detallada acerca de las características de las fuentes de tráfico para datos se encuentra en el Anexo B – sección B.1.

En [15,16], se propone la representación matemática del tráfico de datos usando los modelos MMPP y ON/OFF respectivamente. En [12, 14] se aplican estos conceptos para modelar el tráfico de paquetes a través de una red LAN ó para representar tráfico auto similar y analizar el impacto de este en el tamaño de las colas. En general, el modelamiento de cualquier tráfico de datos exige el uso de una metodología rigurosa que consiste en caracterizar una muestra de tráfico real y representarlo mediante un modelo matemático. La metodología para hacerlo se explica en [12] y se resume en el Anexo B – sección B.2.

El modelo MMPP sugiere el uso de una cadena Markov de dos estados en donde el tiempo de permanencia en cada estado es una distribución de Poisson, es decir exponencial. A diferencia de este, el modelo ON/OFF considera el tiempo de permanencia en cada estado como una distribución de cola pesada, lo cual se ajusta mejor al comportamiento del tráfico IP.

En [16], el modelo de tráfico ON/OFF (ver figura 2.2) implica la representación del tráfico IP mediante un período de inactividad (OFF), seguido de un período de actividad (ON) que consiste en una serie de solicitudes en segundo plano.

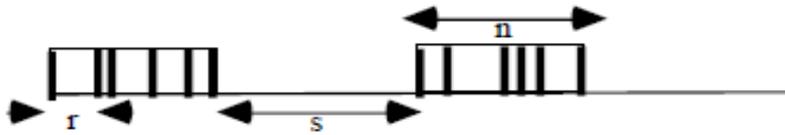


Figura 2.2 Modelo ON/OFF, Fuente [16]

El modelo ON/OFF es descrito como la distribución de tres variables aleatorias:

- r : el tiempo entre arribo de solicitudes durante el estado ON
- s : la duración del período OFF
- n : la duración del período ON

En particular, en [16] se propone usar las siguientes distribuciones estadísticas:

Para el período ON: recomienda usar una distribución Weibull, cuya función de distribución de probabilidad de la variable aleatoria X esta dada por:

$$P[X \leq x] = 1 - e^{-\left(\frac{x}{\beta}\right)^\alpha} \quad (2.3)$$

En donde $x \geq 0$ y $\alpha, \beta \geq 0$. Tanto α y β son respectivamente los parámetros de forma y escala de la distribución.

Para el período OFF: recomienda usar una distribución Pareto, cuya función de distribución de probabilidad de la variable aleatoria X esta dada por:

$$P[X \leq x] = 1 - \left(\frac{k}{x}\right)^\alpha \quad (2.4)$$

En donde k representa el valor más pequeño que puede tomar la variable aleatoria.

Para el tiempo entre arribo: recomienda usar una distribución Weibull

2.1.3 Clases de QoS en las IMT-2000

Las redes celulares IMT-2000 (como 3G y más allá de 3G) se caracterizan por el transporte de tráfico con distintas velocidades binarias. En su mayoría, el tráfico proviene de aplicaciones multimedia que demandan distintos requisitos de QoS. En ese sentido, el protocolo MAC debe ser capaz de atender eficientemente las demandas en cuanto a retardo máximo permitido, ancho de banda mínimo y pérdida de paquetes. En el Anexo C, sección C.1 se abordan las métricas de desempeño para ofrecer QoS y la sección C.2 sugiere el dimensionamiento a tomar en cuenta para determinado tipo de aplicaciones.

De otra parte, la recomendación [19] detalla cuatro clases de QoS para las IMT-2000 desde la perspectiva del usuario:

- clase de servicio conversacional
- clase de servicio interactiva
- clase de servicio unidireccional
- clase de servicio de fondo

La Tabla 2.1 resume las características fundamentales y da un ejemplo para cada una de las cuatro clases definidas.

El factor principal que distingue estas clases entre sí es la sensibilidad de la aplicación al retardo: la clase conversacional se refiere a las aplicaciones muy sensibles al retardo, mientras que la clase de fondo es la clase de QoS menos afectada por el retardo.

Se debe observar que para cualquier aplicación particular es posible que se requiera más de una clase de servicio de QoS

.Para complementar la Tabla 2.2 resume los principales grupos de aplicaciones expresadas por sus requisitos de QoS. La información detallada se encuentra en la referencia [19].

Tabla 2.1

Clases de QoS para las IMT-2000, Fuente [19]

Clase de Servicio de QoS	Clase de Servicio Conversacional	Clase de Servicio Interactiva	Clase de Servicio Unidireccional	Clase de Servicio de Fondo
	Conversación en tiempo real	Clase interactiva al mejor nivel posible	Unidireccional en tiempo real	Clase de fondo al mejor nivel posible
Características fundamentales desde la perspectiva del usuario	Preserva la relación (variación) de tiempo entre las entidades de información del tren transmitido. Modelo conversacional (retardo estricto y bajo)	Patrón de respuestas a peticiones Preserva el contenido útil transportado	Preserva la relación (variación) de tiempo entre las entidades de información del tren Tren continuo unidireccional	El destino no espera recibir los datos dentro de un tiempo determinado Preserva el contenido útil transportado
Ejemplo de la aplicación	voz	Búsqueda en la web	Video unidireccional	Telecarga de correo electrónico

Tabla 2.2

Grupos de comportamiento de aplicaciones según las necesidades de QoS, Fuente [19]

Aplicación	Conversacional (retardo << 1s)	Interactiva (retardo aproximado 1s)	Unidireccional (retardo < 10s)	De fondo (retardo > 10s)
Tolerante al error	Voz y video conversacional	Mensajería de voz	Audio y video unidireccional	fax
Intolerante al error	Telnet, juegos interactivos	Comercio electrónico, búsqueda web, acceso a email	FTP, imagen fija, radio búsqueda	Usenet

2.1.4 Simulador de Eventos Discretos OMNET++

OMNeT++ , [3], es un simulador de eventos discretos, modular y orientado a objetos, cuya aplicación es para redes de comunicaciones. Este simulador es usado para modelar tráfico en redes de telecomunicaciones, modelar protocolos, modelar colas; en general para modelar cualquier sistema donde se pueda aplicar modelado de eventos discretos.

El modelo de OMNeT++ consiste de módulos anidados jerárquicamente sin límite de anidamiento, además los módulos se comunican entre ellos mediante el envío y recepción de “mensajes”. Por otro lado, los mensajes pueden estar constituidos por estructuras de datos complejas y pueden ser enviados al destino ya sea directamente o a través de una ruta predefinida compuesta por puertas y conexiones.

Los módulos tienen sus propios parámetros y los parámetros son usados para personalizar la conducta del módulo y su topología. Los módulos ubicados en el nivel más bajo de la jerarquía encapsulan el comportamiento y son llamados “módulos simples”. Justamente los módulos simples son programados en lenguaje C++ y se integran a la librería de simulación de OMNET++ (API Simulation Library). Los detalles del simulador OMNET++ se detallan en el Anexo D.

2.1.4.1 Los archivos requeridos por OMNET++

En esta tesis se ha programado un protocolo MAC usando el simulador de eventos discretos OMNET++. Una simulación en OMNET++ requiere contar con los siguientes archivos:

- El archivo ó archivos con extensión *.cc
- El archivo network.ned
- El archivo omnetpp.ini
- El archivo ó archivos con extensión *.msg

Los archivos con extensión *.cc, vienen a ser programas en C++ y definen la conducta de los módulos simples. Se debe recordar que un módulo compuesto se

compone de uno o más módulos simples. Cabe anotar que la mayor parte del programa para protocolo MAC se definió en estos archivos.

El archivo `network.ned` define la topología del ambiente de simulación. Aquí se define como se enlazan los módulos simples para formar un módulo compuesto, como se interconectan los módulos compuestos. También se definen las puertas de enlace entre módulos, como se enlazan los archivos de extensión `*.cc` con el archivo `network.ned` y también se definen los parámetros y enlaces al archivo `omnetpp.ini`.

Dentro del archivo `omnetpp.ini` se debe especificar el valor de los parámetros que no se definieron dentro de los archivos `*.cc` ó dentro del archivo `network.ned`; también se especifican las opciones para la simulación como: tiempo de duración de la simulación, ambiente de ejecución (`Tkenv` ó `Cmdenv` – ver Anexo D), y nombre de la simulación entre otras cosas adicionales.

El archivo con extensión `*.msg`, define los mensajes que se cursarán entre los módulos, cada vez que se ejecuta un evento. Este tipo de archivo(s) es opcional a la simulación y suele definirse cuando el programador desea personalizar el tipo de mensajes que se cursarán entre los módulos. Todos los mensajes nuevos que se definan serán una subclase de la clase básica `cMessage`. Para el caso particular de esta tesis se tuvo que crear una instancia de `cMessage` exclusiva para el protocolo MAC eficiente.

Es importante indicar que durante y al final de la simulación se generarán los archivos de estadística y que sirven para evaluar el desempeño del protocolo MAC. Estos archivos son `omnetpp.vec` y `omnetpp.sca`, el primero almacena las estadísticas vectoriales mientras que el segundo almacena las estadísticas escalares. Los datos que generan ambos archivos vienen en formato de texto y requieren un procesamiento posterior.

Al término de la programación y definición de los archivos mencionados anteriormente, se debe proceder a compilarlos y enlazarlos con el programa de simulación principal; esta operación crea un programa ejecutable. Cuando se corre el programa ejecutable, este se enlaza al archivo omnetpp.ini y al término de la simulación se generan los archivos estadísticos. En la figura 2.3 se indican los puntos dentro del framework que fueron modificados y personalizados para desarrollar el protocolo MAC eficiente de esta tesis.

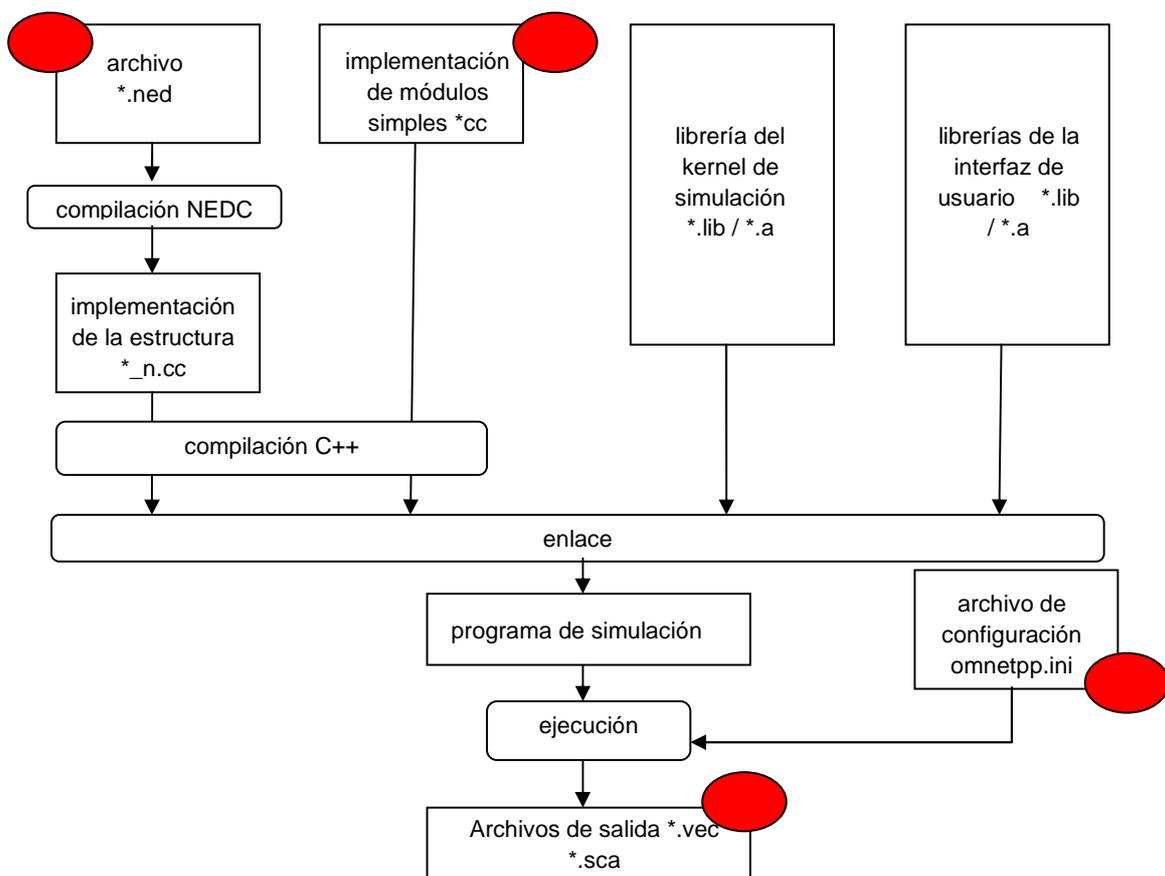


Figura 2.3 Enlace de archivos dentro de OMNET++, Fuente [3], OMNET++ Manual año 2005

2.2 Diseño del protocolo MAC

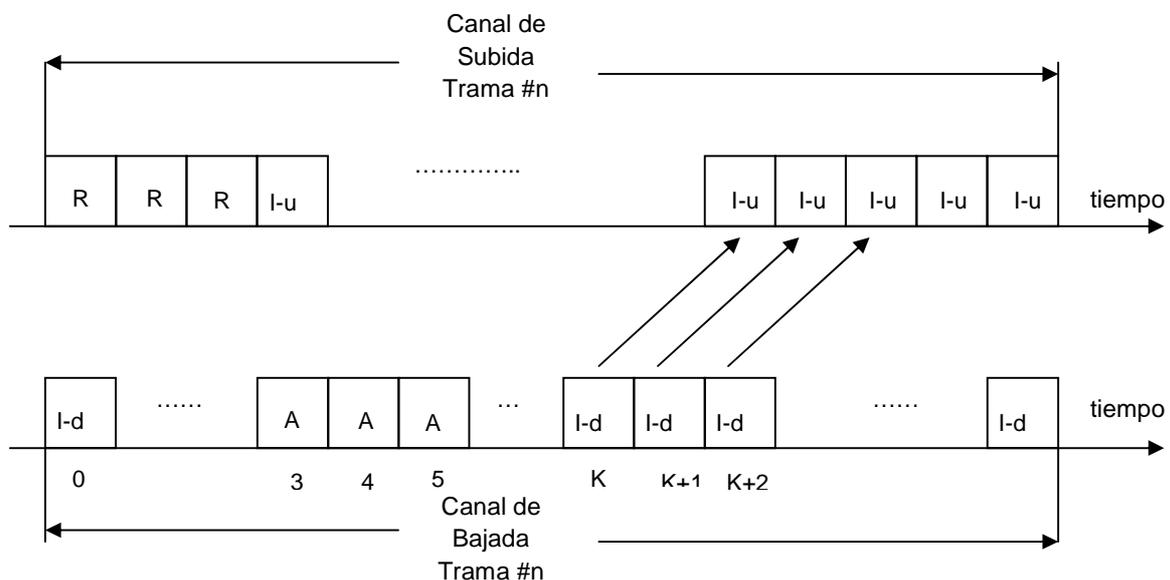
Como se comentó en la sección 1.4 (Solución Planteada), el protocolo MAC de esta tesis se construyó tomando como base de diseño el protocolo CRDA [2], al cual se realizaron las modificaciones necesarias, a fin de que soporte el transporte de tráfico con diferente QoS y responda eficientemente a los distintos tipos de tráfico.

Portar el protocolo CRDA a OMNET++ no fue una tarea trivial porque no se contaba con el código fuente del protocolo. Para poder realizar esta tarea, fue necesario estudiar detalladamente el protocolo CRDA, luego traducir su estructura de trama y funcionamiento en modelos lógicos y fórmulas de diseño, que permitan representarlo claramente. Los modelos lógicos están conformados por diagramas de flujo del funcionamiento del protocolo, diagramas de señalización y temporizadores del protocolo. De otra parte, para deducir las fórmulas de diseño primero fue indispensable determinar los parámetros que definen la estructura del protocolo. Para completar la información del diseño, también se detallan las estructuras internas de las ranuras que conforman las tramas de los canales de subida y bajada.

Las modificaciones realizadas a [2], también se ven reflejadas en el diseño del protocolo y fueron básicamente dos. La primera consistió en considerar como fuentes de tráfico a cuatro aplicaciones con distinta demanda de QoS. Las aplicaciones consideradas fueron: juegos en línea, búsqueda web, video streaming y datos sin prioridad. La segunda consistió en la implementación de una política para priorización de tráfico, en el algoritmo de planificación de la estación base.

2.2.1 Estructura de la Trama

El protocolo MAC de esta tesis, utiliza la estructura expuesta en [2], la que consiste en una trama TDMA con 72 ranuras (slots), tanto para el canal de subida como para el canal de bajada. La Figura 2.4 corresponde a la trama del protocolo CRDA y que aplica también al protocolo MAC de esta tesis.



Leyenda:

R : ranura de reserva ó R-slot

A : ranura de aceptación ó A-slot

I : ranura de información:

I-u ó I-slot-up (si pertenece al canal de subida), ó

I-d ó I-slot-down (si pertenece al canal de bajada)

Figura 2.4 Estructura de la trama TDMA del Protocolo MAC Eficiente, Fuente [2]

2.2.1.1 Parámetros de Diseño

Para la estructura de trama de la figura 2.4, en esta sección se listan los parámetros necesarios para el diseño de las tramas de los canales de subida y bajada.

Parámetros para el canal de subida

R-slot : Ranura de Reserva

A través de esta ranura los terminales móviles envían las solicitudes de reserva de canales de tráfico hacia la estación base.

I-slot-up : Ranura de Información para el canal de subida

A través de esta ranura los terminales móviles envían el tráfico de voz o datos hacia la estación base.

#I-slot-up : Cantidad de I-slot-up

Cantidad de ranuras de información para el canal de subida. Para nuestro caso el valor de este parámetro es 68.

R-slot-duration : Duración en tiempo del R-slot

Indica la duración en tiempo de una ranura de reserva. Su valor se deduce usando la fórmula 2.5. Para nuestro caso usamos un valor de $n=4$.

I-slot-up-duration : Duración en tiempo del I-slot-up

Indica la duración en tiempo de una ranura de información de subida. Su valor se infiere de la fórmula 2.6.

Frame-up-duration : Duración en tiempo de una trama de subida del protocolo MAC

Indica la duración en tiempo de una trama TDMA de subida. Para nuestro caso el valor usado es 5 mseg ([21]).

Parámetros para el canal de bajada

A-slot : Ranura de Aceptación

A través de esta ranura la estación base informa a los terminales móviles sobre la aceptación de las solicitudes de reserva enviadas por el R-slot.

I-slot-down : Ranura de Información para el canal de bajada

A través de esta ranura la estación base envía el tráfico proveniente de la red hacia los terminales móviles. También por estas ranuras se notifica a las terminales móviles, que tienen tráfico pendiente de envío, sobre la posición de la ranura de información en donde han ganado el permiso para enviar.

#I-slot-down : Cantidad de I-slot-down

Cantidad de ranuras de información para el canal de bajada. Para nuestro caso el valor de este parámetro es 68.

A-slot-duration : Duración en tiempo del A-slot

Indica la duración en tiempo de una ranura de aceptación. Su valor se deduce usando la fórmula 2.7.

I-slot-down-duration : Duración en tiempo del I-slot-down

Indica la duración en tiempo de una ranura de información de bajada, Su valor se deduce usando la fórmula 2.10.

Frame-down-duration : Duración en tiempo de una trama de bajada del protocolo MAC

Indica la duración en tiempo de una trama TDMA de bajada. Para nuestro caso el valor usado es 5 mseg ([21]).

Parámetros Generales

R/A shift : Desplazamiento entre las ranuras R y A

Para efectos prácticos a este desplazamiento lo llamaremos **k**. Su valor está expresado en la cantidad de ranuras de información que separan el inicio de una ranura R (R-slot) y el inicio de una ranura A. (A-slot) por ejemplo, en la figura 2.5 se representa la sincronización de dos tramas, una de subida y otra de bajada. El R-slot inicia en la ranura de información cero, mientras que el A-slot inicia en la ranura de información tres. El valor de k se deduce de la fórmula 2.12 y se usó un valor de $m=4$.

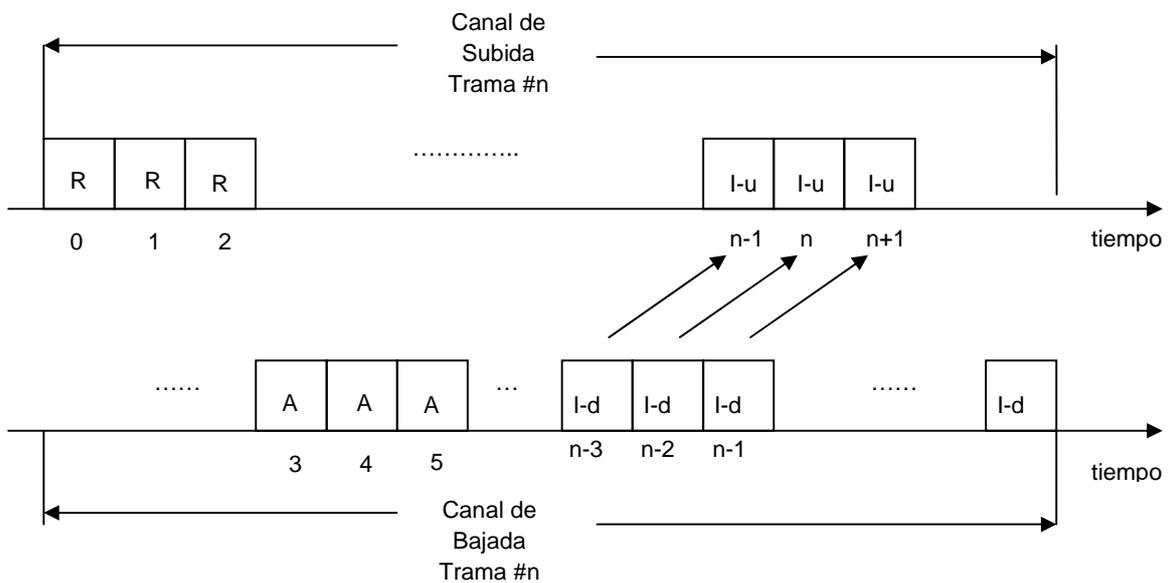


Figura 2.5 Posiciones de los R-, A-, I-slots para dos tramas de subida y bajada sincronizadas, Fuente [2]

En la figura 2.5 cabe agregar la siguiente leyenda:

R : ranura de reserva ó R-slot

A : ranura de aceptación ó A-slot

I : ranura de información:

I-u ó I-slot-up (si pertenece al canal de subida), ó

I-d ó I-slot-down (si pertenece al canal de bajada)

I-slot-offset : Desplazamiento entre una ranura de Información de bajada y una ranura de información de subida, solo para casos de notificación

Para efectos prácticos a este desplazamiento lo llamaremos **h**. El valor se deduce de la fórmula 2.13. El uso de este parámetro viene dado cuando la estación base le concede permiso para transmitir a un terminal móvil. La estación base notifica al terminal móvil a través del A-slot y en el canal de bajada, marca indirectamente las ranuras de información en donde el terminal móvil tendrá permiso para transmitir. De esta manera, al escuchar las ranuras de información de bajada, el terminal móvil sabe de antemano en que ranuras de información de subida tendrá derecho a transmitir ya que añadirá el valor de **h (I-slot-offset)** a la posición de la ranura de información de bajada en donde recibió la notificación.

Como ejemplo, supongamos que la estación base le concede permiso para transmisión a un terminal móvil y le asigna una sola ranura de información del canal de subida. Siguiendo el ejemplo, la estación base informará al terminal móvil acerca de la posición de la ranura de información de subida, en la ranura **x** que pertenece a una ranura de información de bajada. De esta manera el terminal móvil transmitirá en la ranura **x+h**, correspondiente a una ranura de información de subida. Es importante indicar el efecto de frontera (Figura 2.6) que afecta la asignación de I-slots de subida y se origina en la vecindad entre la finalización de una trama y el inicio de la siguiente trama. Aparentemente el valor de **h** podría ser fijo para todas las asignaciones de **I-slot-up** pero en la vecindad de finalización en la trama de bajada, el valor de **h** deberá tener en cuenta el espacio en tiempo del **R-slot**. Con lo expuesto, el algoritmo de asignación residente en el terminal móvil

deberá agregar el valor **k (R/A shift)** al valor de **h (I-slot-offset)** cuando se acerca la finalización de la trama de bajada.

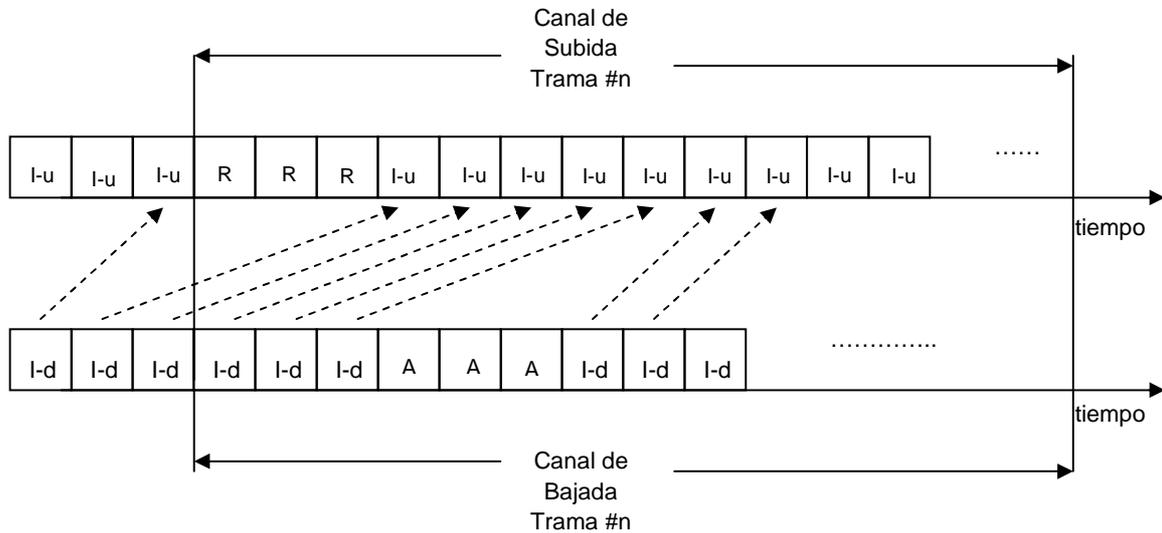


Figura 2.6 Efecto de frontera en la asignación de I-slot-up, Fuente [2]

En la figura 2.6 cabe agregar la siguiente leyenda:

R : ranura de reserva ó R-slot

A : ranura de aceptación ó A-slot

I : ranura de información:

I-u ó I-slot-up (si pertenece al canal de subida), ó

I-d ó I-slot-down (si pertenece al canal de bajada)

bwt : Temporizador para el retardo

Este valor representa el retardo máximo tolerado por la terminal móvil, que acaba de enviar una solicitud de reserva, antes de que se le notifique sobre la disponibilidad de I-slots-up. Cuando el temporizador bwt expira, el terminal móvil elimina los paquetes de la cola local que están pendientes de envío. Ver fórmula 2.14.

2.2.1.2 Fórmulas de Diseño

Habiendo inferido los parámetros de diseño de la trama del protocolo MAC, ahora se enuncian las fórmulas que establecen las relaciones y valores de los parámetros.

Fórmulas para la Trama de Subida

$$R\text{-slot-duration} = n * I\text{-slot-up-duration} \quad (2.5)$$

; donde n es un entero (valor operacional)

$$\text{Frame-up-duration} = R\text{-slot-duration} + (\#I\text{-slot-up} * I\text{-slot-up-duration}) \quad (2.6)$$

Fórmulas para la Trama de Bajada

$$A\text{-slot-duration} = R\text{-slot-duration} \quad (2.7)$$

$$\text{Frame-down-duration} = A\text{-slot-duration} + (\#I\text{-slot-down} * I\text{-slot-down-duration}) \quad (2.8)$$

Fórmulas Generales

$$\text{Frame-up-duration} = \text{Frame-down-duration} \quad (2.9)$$

$$I\text{-slot-up-duration} = I\text{-slot-down-duration} \quad (2.10)$$

$$\#I\text{-slot-up} = \#I\text{-slot-down} \quad (2.11)$$

$$k = m * I\text{-slot-up-duration} \quad (2.12)$$

; donde m es un entero (valor operacional)

$$h = \begin{cases} 2; & \text{para las posiciones de I-slot-down desde 01 al 63} \\ 2 + R\text{-slot-duration}; & \text{para las posiciones de I-slot-down desde 64 al 68} \end{cases} \quad (2.13)$$

$$bwt = \begin{cases} \text{Valor especificado en la simulación;} & \text{para voz} \\ \text{Valor especificado en la simulación;} & \text{para datos} \end{cases}$$

(2.14)

2.2.1.3 Estructura Interna de las Ranuras

Para complementar la información del diseño de la trama, en esta sección se expone la estructura interna ó formato de las ranuras: R-slot, A-slot, I-slot-up, I-slot-down. La figura 2.7 resume esta sección.

Estructura de la Ranura de Reserva (R-slot)

Una ranura de reserva contiene información como:

- Dirección MAC del terminal móvil
- Código CDMA asignado al terminal móvil
- Tipo de servicio (TOS), usado por el terminal móvil para especificar la QoS del tráfico que se desea enviar
- Requerimiento de Recursos, usado por el terminal móvil para informar a la estación base acerca de la cantidad de ranuras de información necesarias para transmitir todos los paquetes en la cola local.

A esta estructura también se le llamará SIG.

Estructura de la Ranura de Información para el canal de Subida (I-slot-up)

La estructura de esta ranura se compone de una cabecera y una carga útil.

La cabecera contienen información como:

- Dirección MAC del terminal móvil
- Contador de longitud de mensajes, usado para informar a la estación base acerca del tamaño de los mensajes (datos) almacenados en la cola local del terminal móvil

- Bit para Piggybacking, usado para informar a la estación base acerca del arribo de mensajes (datos) adicionales en la cola local.

La carga útil es usada para el transporte de la información.

Estructura de la Ranura de Aceptación (A-slot)

Una ranura de aceptación se compone por:

- Dirección MAC del terminal móvil que transmitió la solicitud de reserva (SIG)
- Código CDMA asignado al terminal móvil que transmitió la solicitud de reserva (SIG)
- Campo de ACK, usado para notificar a los terminales móviles con acceso al canal de subida.

A esta estructura también se le llamará ACK.

ESTRUCTURA DE LA RANURA DE RESERVA (R-SLOT)

Dirección MAC del terminal móvil	Código CDMA asignado al terminal móvil (ON)	Tipo de servicio - QOS	Requerimiento de Recursos
----------------------------------	---	------------------------	---------------------------

ESTRUCTURA DE LA RANURA DE ACEPTACION (A-SLOT)

Dirección MAC del terminal móvil	Código CDMA asignado al terminal móvil	ACK
----------------------------------	--	-----

ESTRUCTURA DE LA RANURA DE INFORMACIÓN DE SUBIDA (I-SLOT-UP)

CABECERA	CARGA	
Dirección MAC del terminal móvil	Bit de piggybacking	Contador de longitud de mensajes

ESTRUCTURA DE LA RANURA DE INFORMACIÓN DE BAJADA (I-SLOT-DOWN)

CABECERA	CARGA	
Dirección MAC del terminal móvil	Bit de estado	Dirección MAC del terminal exitoso

Figura 2.7 Estructura Interna de las Ranuras

Estructura de la Ranura de Información para el canal de Bajada (I-slot-down)

Una ranura I-slot-down se compone de una carga útil y una cabecera

La cabecera contiene información como:

- Dirección MAC del terminal móvil a quien va dirigido el tráfico de bajada que se transporta
- Bit de estado: cuando el bit está desactivado entonces inhabilita el campo de "Dirección MAC de terminal exitoso"
- Dirección MAC de terminal exitoso, como su nombre lo indica, este campo lleva la dirección MAC del terminal con permiso para transmitir en el canal de subida (terminal exitoso)

2.2.2 Fuentes de Tráfico Utilizadas

El protocolo MAC de esta tesis utiliza modelos matemáticos de fuentes de tráfico para las siguientes aplicaciones:

- Aplicación de voz con VAD activado
- Aplicación de juegos en línea
- Aplicación de búsqueda web
- Aplicación de video streaming
- Aplicación de datos sin prioridad

Estos modelos habilitan la operación de los generadores de fuentes de tráfico, que se ubican en las terminales móviles, a fin de enviar paquetes de datos hacia la estación base.

Las aplicaciones mencionadas (a excepción de la voz) se basan en los modelos ON/OFF que se explicaron en la sección 2.1.2.2 pero con modificaciones específicas en el tipo de distribución estadística utilizado.

Las aplicaciones seleccionadas cumplen con los lineamientos de la recomendación [19] y sus valores fueron sacados de las referencias [20], [22] y [2].

Aplicación de Voz con VAD activado

Utiliza el modelo de tráfico de voz de la sección 2.1.2.1 (Brady). Los valores de los parámetros se indican en la Tabla 2.4

Aplicación de Juegos en línea

Basado en el modelo ON/OFF, emula el tráfico conversacional. En este modelo se define el proceso de arribo de un paquete y dentro de la llegada de un paquete se presenta el proceso de arribo de datagramas. Dentro de este modelo, el proceso de arribo de las sesiones por paquete no se especifica y se debe asumir como continuo durante todo el proceso de la simulación; es decir una sesión equivale al tiempo total de la simulación.

Para el proceso de arribo de paquetes se debe especificar el tiempo de duración del paquete (tiempo ON) y el tiempo de lectura (tiempo OFF). El tiempo de lectura se inicia luego de la transmisión exitosa de todos los datagramas generados durante el arribo del paquete previo. Para el proceso de arribo de los datagramas se debe especificar el tamaño del paquete (en bits) y el tiempo entre arribo de los datagramas. La situación anterior se esquematiza en la figura 2.8.

Los parámetros de simulación usados para este modelo fueron sacados de [20] y sus valores se detallan en la Tabla 2.6:

- Distribución ON (Distribución para la llegada de paquetes): Exponencial
- Distribución OFF (Distribución para el tiempo de lectura): Exponencial
- Distribución del Tiempo entre arribo (de datagramas): Normal logarítmico
- Tamaño de Datagrama: Fijo

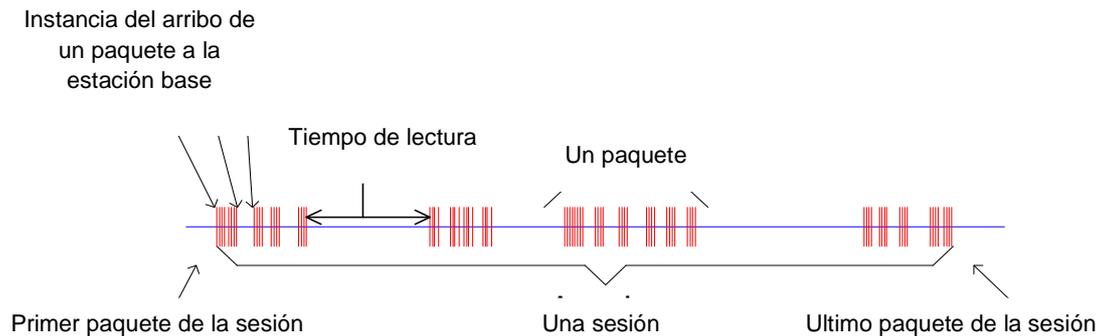


Figura 2.8 Modelo para una Fuente de tráfico Interactiva, Fuente [20]

Aplicación de Búsqueda web

Basado en el modelo ON/OFF, emula el tráfico interactivo. Este modelo consta de tres niveles:

- Tráfico del nivel de sesión
- Tráfico del nivel de conexión
- Tráfico del nivel de paquete

En cada sesión se genera un determinado número de conexiones y cada conexión genera un número determinado de paquetes. En nuestras simulaciones asumimos una sola sesión con una duración equivalente al tiempo de simulación.

Los parámetros de simulación usados para este modelo fueron sacados de [22] y sus valores se detallan en la Tabla 2.7 (para nuestras simulaciones se usaron los valores correspondientes a los modelos de 64Kbps):

- Distribución ON (Distribución para el nivel de conexión): Lognormal.
- Distribución OFF: no se considera
- Distribución del Tiempo entre arribo (Distribución para el nivel de paquetes): Lognormal
- Volumen de Datos Lognormal

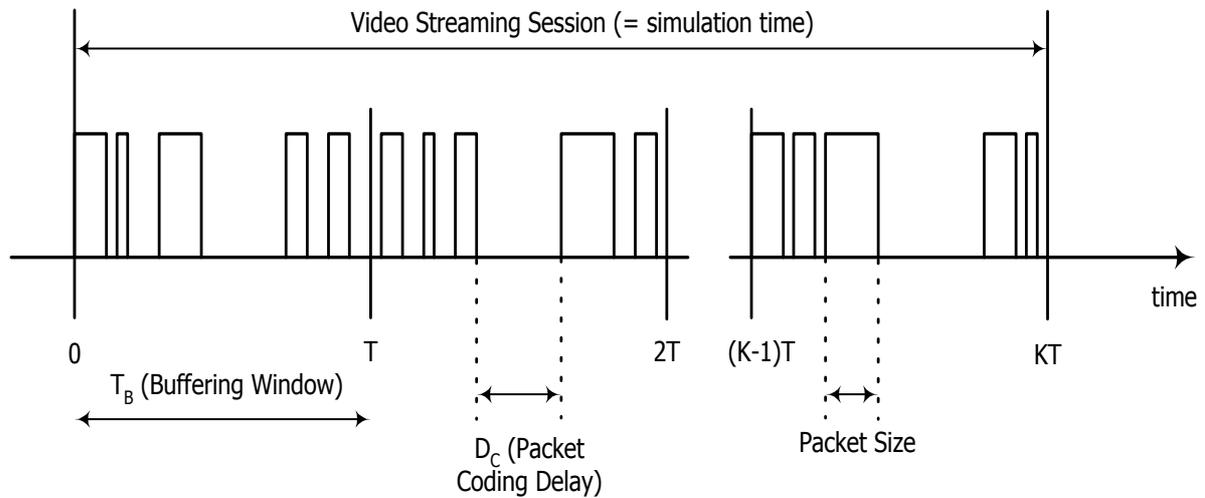
- Tamaño del paquete generado: Fijo (576 bytes)

Aplicación de Video Streaming

Basado en el modelo ON/OFF, emula el tráfico unidireccional. La figura 2.9 esquematiza el modelo de tráfico para flujo de video (video streaming). Este modelo no considera el retardo de latencia originado al inicio de la llamada de video. Una sesión que contiene flujo de video se define como una llamada completa de video, cuyo tiempo es igual a la duración total de la simulación. Cada marco que contiene datos de video, llega a intervalos regulares, definidos por **T** y por el número de marcos por segundo (frames per second - fps). Cada marco se descompone en un número fijo de partes, y cada parte es transmitida como si fuese un solo paquete. El tamaño de dichos paquetes/partes es modelado a través de una distribución truncada de Pareto. Por otro lado, el retardo por codificación (**Dc**) originado por el codificador de video, introduce intervalo de retardo entre los paquetes de un marco. Estos intervalos también son modelados por medio de una distribución truncada de Pareto.

También se define el parámetro **TB** que es la longitud (en segundos) del buffer anti retardo (en inglés de-jitter buffer window) y localizado dentro de la red, cuya finalidad es garantizar la continuidad en el flujo de datos de video. Este parámetro no es relevante para la generación de las distribuciones de tráfico pero es de gran ayuda para identificar los períodos en donde no se alcanzan las expectativas mínimas del servicio.

Los parámetros de simulación usados para este modelo se fueron sacados de [20] y sus valores se detallan en la Tabla 2.8:



Leyenda:

T_B : buffer anti retardo

D_C : retardo por codificación

T : período

K : cantidad de períodos de una sesión

Packet size tamaño del paquete

Video streaming session : sesión de flujo de video unidireccional

Figura 2.9 Modelo de tráfico para flujo de video, Fuente [20]

Aplicación de datos sin prioridad

Basado en el modelo ON/OFF, emula el tráfico de fondo. Para efectos prácticos se asume que, los parámetros de simulación usados para este modelo son los que se especifican en [2] como modelo de tráfico de datos:

Se modela una fuente generadora de tráfico de datos con las siguientes características:

- Distribución ON: Weibull
- Distribución OFF: Weibull
- Distribución del Tiempo entre Arribo: Exponencial
- Longitud de Mensajes: Fijo

2.2.3 Funcionamiento del Protocolo

El funcionamiento del protocolo MAC de esta tesis se basa en el protocolo CRDA [2], con una modificación en el algoritmo de planificación de paquetes que reside en la estación base. Dicha modificación se hizo a fin de habilitar el soporte de QoS. El aporte en esta sección es proveer los diagramas de flujo que describen el funcionamiento de CRDA y que son también parte del diseño del protocolo de esta tesis. El funcionamiento del protocolo CRDA varía en función del tráfico solicitante: si es voz o es datos. Cuando se genera tráfico de voz se habla de “paquetes”, cuando se genera tráfico de datos se habla de “mensajes”. En ese sentido se está diferenciando el tipo de tráfico.

El protocolo MAC de esta tesis mantiene la misma diferenciación, pero además, dentro del grupo de datos, añade un subgrupo para diferenciar el tipo de aplicación de datos. Este subgrupo clasifica el tráfico de datos en conversacional, interactivo, unidireccional y de fondo. Las aplicaciones que encajan con cada tipo de tráfico ya fueron expuestas en la sección 2.2.2. De esta manera cuando se genera un mensaje, este puede deberse a una aplicación de juegos en línea, búsqueda web, video streaming ó de datos sin prioridad.

Cuando se genera un paquete (de voz) ó un mensaje (de datos), estos pueden contener una ó varias celda(s). Una celda equivale a un slot de tiempo de la trama TDMA; en este caso un i-slot-up. Además un paquete equivale a una celda mientras un mensaje puede ser equivalente a una cantidad finita de celdas. Cuando se genera tráfico en el terminal móvil, este transforma el paquete ó mensaje, en una cantidad finita de celdas que luego son enviadas a la cola local para esperar su posterior envío a la estación base.

Cuando la cola local tiene celdas por enviar, se activa el algoritmo de reserva, que se encarga de realizar la petición de transmisión de las celdas. Una vez aceptada la petición, el terminal móvil entrará en una fase de administración de ancho de banda que finalizará cuando la cola local haya sido vaciada por completo.

Del lado de la estación base, el algoritmo de planificación de paquetes/mensajes (scheduling algorithm) se encargará de administrar y asignar las ranuras del canal de subida (I-slot-up) a los terminales móviles que tengan celdas por transmitir.

En la Figura 2.10 se ilustra la ubicación de los algoritmos de reserva y planificación, en el terminal móvil y estación base respectivamente.

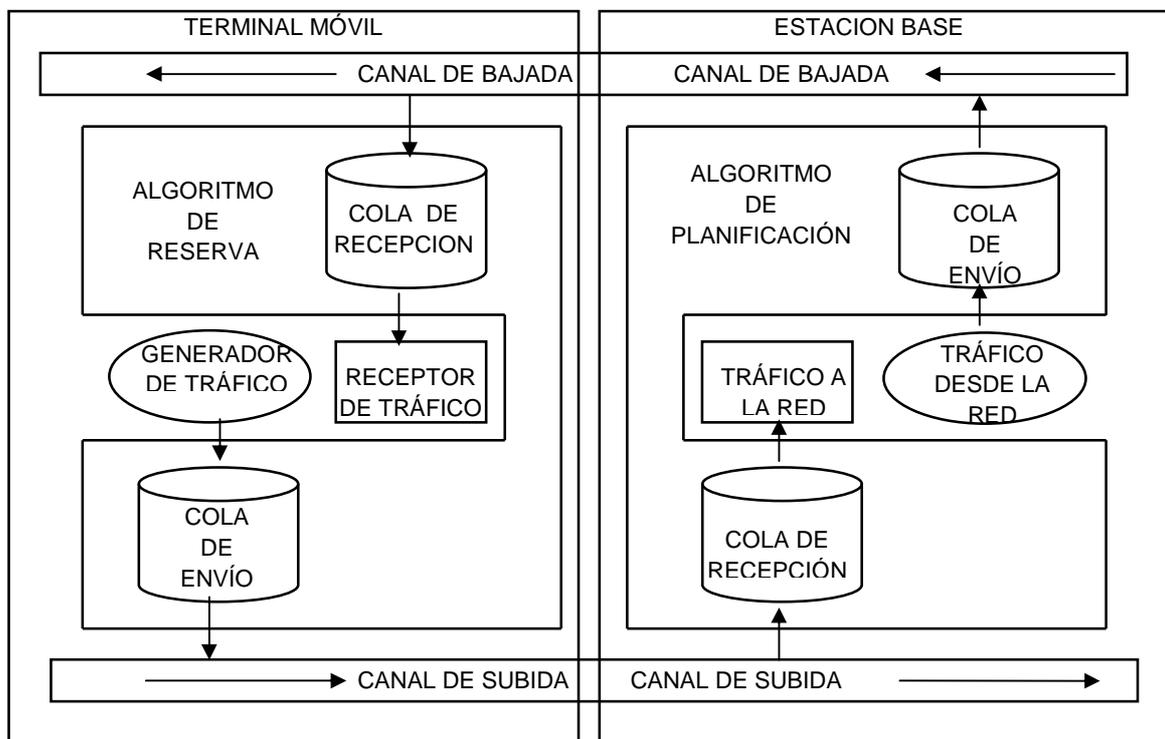


Figura 2.10 Esquema con los Algoritmos residentes en cada Objeto (Terminal Móvil y Estación Base)

Al inicio de la simulación todos los generadores de tráfico se encuentran apagados (OFF) y aleatoriamente van pasando al estado de encendido (ON). Cuando esto sucede, se pueden enviar una o más celdas hacia la cola local; de esta forma el terminal móvil cambia su estado a “terminal móvil esperando”. Inmediatamente el terminal móvil entra en una fase de reserva de canal y espera la llegada del R-slot,

cuando esto sucede envía una petición de transmisión llamada SIG a través del R-slot. Dicho SIG es recibido por la estación base provocando el inicio de la fase de planificación de paquetes/mensajes. Seguidamente, la estación base responde vía el A-slot, enviando una señal de aceptación, llamada ACK, al terminal móvil y notificando que la “petición se encuentra en trámite”. Inmediatamente la estación base busca si existen slots (I-slot-up) disponibles, y si lo(s) encuentra, entonces se asigna(n) el (los) I-slot-up al terminal móvil. La notificación sobre la posición del I-slot-up se realiza a través del I-slot-down, dicha notificación indica que la “petición fue aceptada y asignada”. Seguidamente el terminal móvil recibe la notificación y cambia su estado a “terminal móvil exitoso”. El diagrama de flujo de la figura 2.11 generaliza la situación explicada. Los detalles concernientes a cada fase se explican a continuación pues dependen del tipo de tráfico.

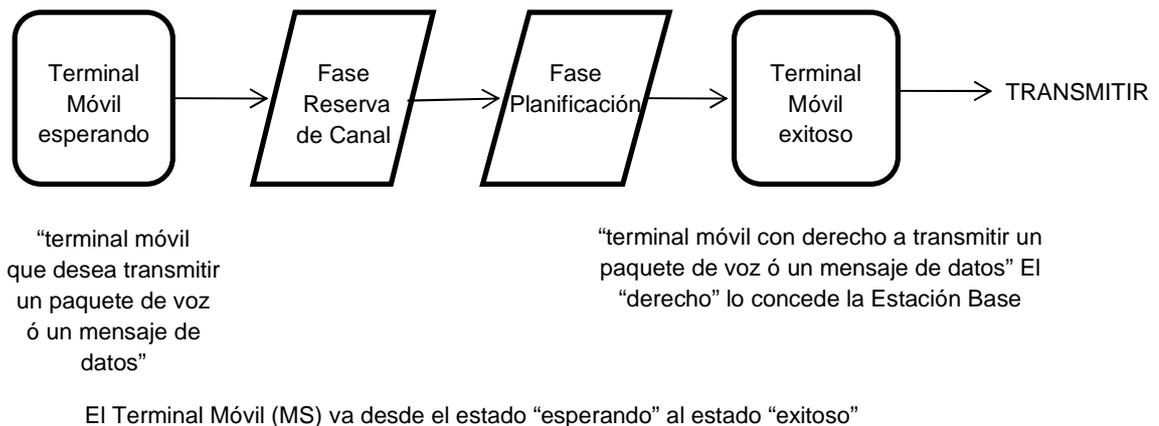


Figura 2.11 Evolución del estado en el terminal móvil

De aquí en adelante se mostrarán distintos diagramas de flujo que se enlazan mediante conectores (en numeración romana y dentro de un círculo).

2.2.3.1 Funcionamiento para Voz

Para el caso del terminal móvil de voz, el funcionamiento del protocolo MAC está diseñado para emular las características de un circuito conmutado, esto significa que durante el tiempo que el generador de tráfico de voz se mantenga activo, el recurso de i-slot-up se quedará asignado al terminal móvil hasta que el generador de tráfico se apague. Si por alguna razón el terminal móvil no recibe la asignación de un i-slot-up durante un tiempo determinado, los paquetes de voz (celdas contenidas en la cola local) comenzarán a ser eliminados por el terminal móvil originándose pérdida de paquetes. Cuando se trata de voz, la estación base asignará únicamente un i-slot-up al terminal móvil.

El funcionamiento del protocolo MAC de esta tesis, en el caso de voz, es idéntico al que se explica en [2]. Por ello en esta sección solo se proveen una breve descripción y los diagramas de flujo que ayudaron a la implementación del protocolo. Para mayor detalle se sugiere revisar la referencia [2].

2.2.3.1.1 Fase de Reserva de Canal

Esta fase es controlada por el algoritmo de reserva del terminal móvil. Los diagramas de flujo que simbolizan el comportamiento de la fase de reserva se muestra en las figuras 2.12, 2.13 y 2.14.

Tan pronto se genera el tráfico de voz, los paquetes son transformados en celdas y van colocándose en la cola local. Inmediatamente el terminal móvil espera la llegada del siguiente R-slot para enviar una señal SIG (sección 2.2.1.3 Estructura de la Ranura de Reserva). El diagrama de flujo se inicia en la figura 2.12 y continúa en la figura 2.13

Seguidamente el terminal móvil monitorea el A-slot para saber si el SIG enviado tuvo éxito. Si falla, el terminal móvil intentará transmitir un nuevo SIG en el siguiente R-slot, caso contrario empieza a monitorear los canales I-slot-down

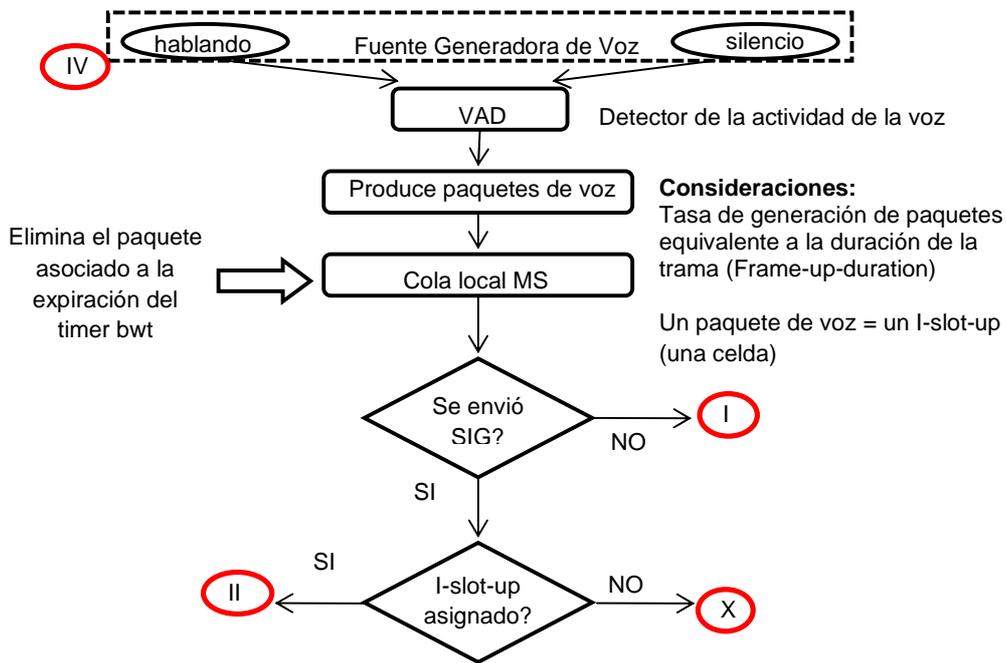


Figura 2.12 Fase de Reserva de Canal – Lado del Terminal Móvil (Funcionamiento para Voz, parte A)

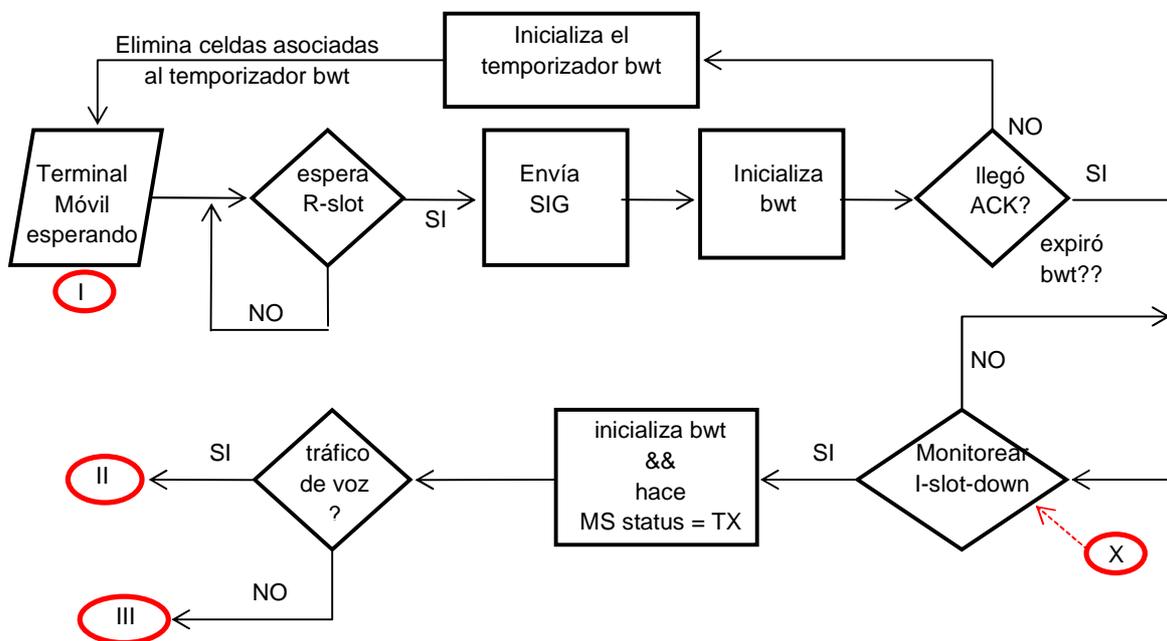


Figura 2.13 Fase de Reserva de Canal – Lado del Terminal Móvil (Funcionamiento para Voz, parte B)

Cuando el terminal móvil se entera del i-slot-up asignado (a través de la notificación que recibió por el i-slot-down), procede a transmitir una celda de la cola local. Luego espera la siguiente trama para seguir transmitiendo por el mismo i-slot-up que le fue asignado al inicio. Este proceso se repite hasta vaciar la cola local. Cuando esto sucede, el terminal móvil libera el i-slot-up asignado y cambia su estado a LIBRE. La figura 2.14 completa esta explicación.

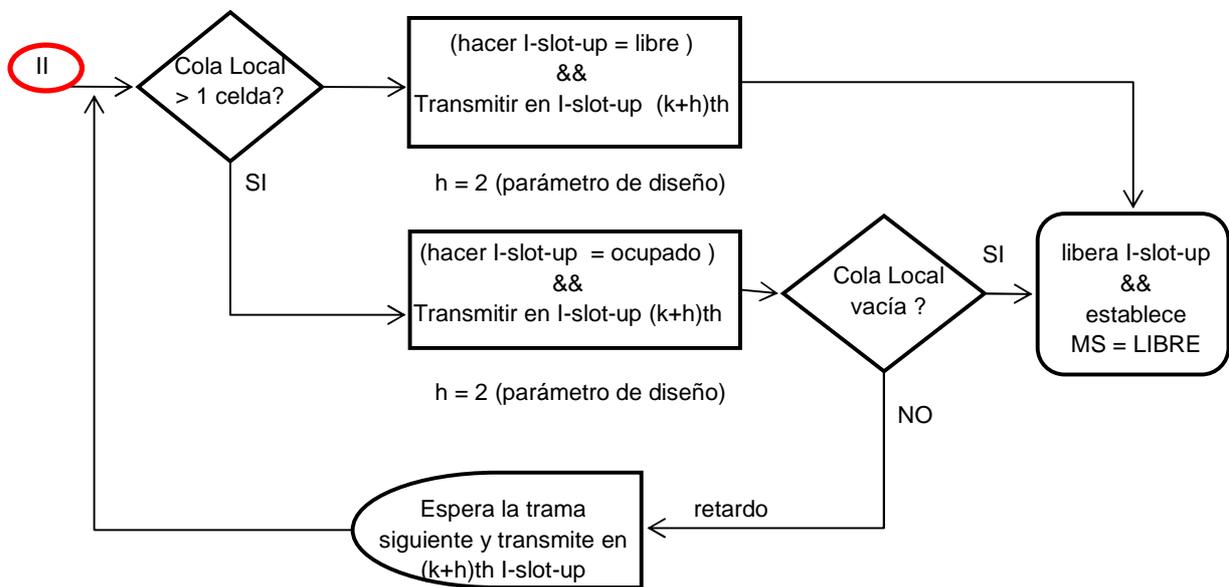


Figura 2.14 Fase de Reserva de Canal – Lado del Terminal Móvil (Funcionamiento para Voz, parte C)

2.2.3.1.2 Fase de Planificación

Esta fase es controlada por el algoritmo de planificación de la estación base. El diagrama de flujo que simboliza el comportamiento de la fase de planificación se muestra en las figuras 2.15.

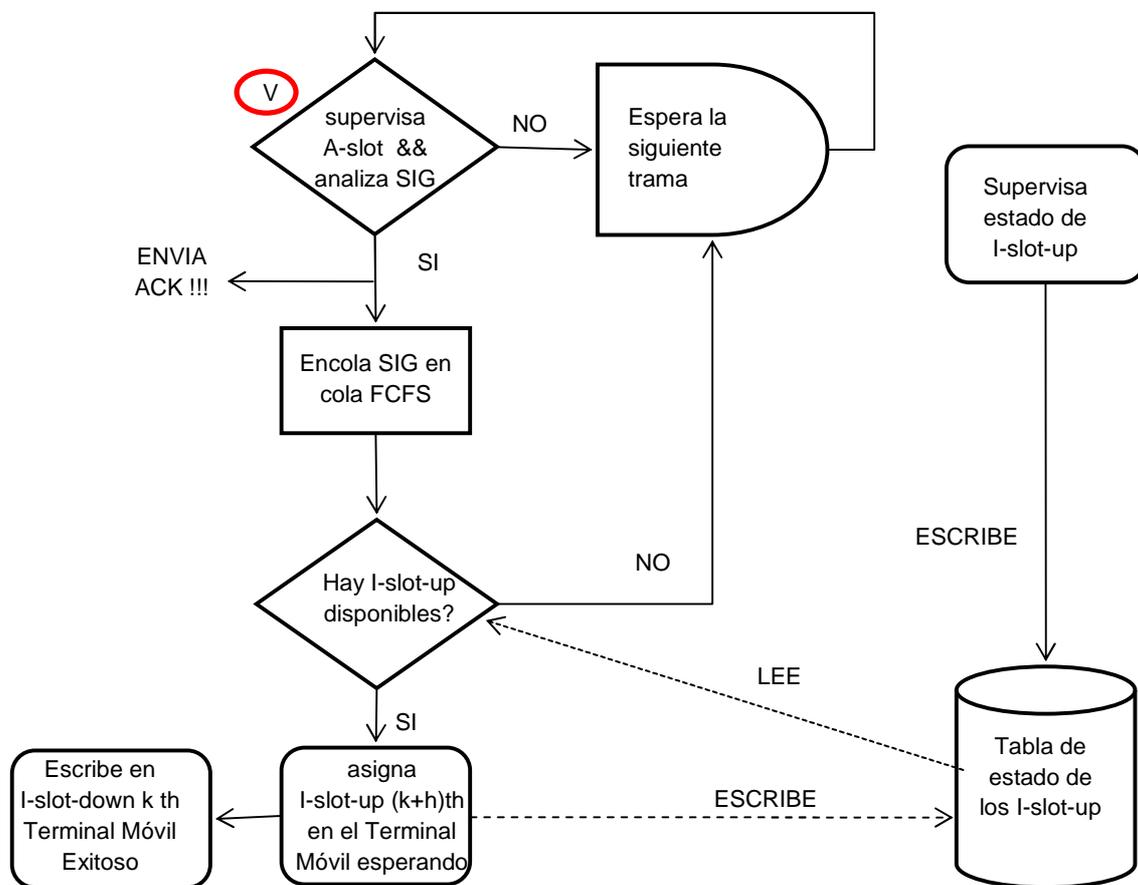


Figura 2.15 Fase de Planificación – Lado de la Estación Base (Funcionamiento para Voz)

Esta fase se inicia con la finalización del R-slot (inicio del A-slot). Inmediatamente la BS analiza si ha recibido SIGs, en caso no tenga ninguno, espera la siguiente trama. En caso contrario va encolando los SIGs recibidos en una cola FCFS y automáticamente va enviando las notificaciones ACK a los terminales móviles. Seguidamente procede a asignar los recursos de i-slot-up y envía las notificaciones a través del canal de bajada escribiendo en los i-slot-down la dirección de los terminales que ya ganaron el permiso para transmitir.

Cabe resaltar que la BS va ir actualizando la tabla de estado de i-slot-up cada vez que recibe tráfico proveniente de los terminales móviles (MS).

2.2.3.2 Funcionamiento para Datos de Velocidad Binaria Variable

En el caso de datos, el funcionamiento del protocolo MAC está diseñado para emular las características de la transmisión de paquetes por demanda, esto significa que los datos se transmiten por ráfagas y a una velocidad controlada por el parámetro de calidad de servicio (QoS).

El generador de tráfico residente en el terminal móvil genera mensajes de datos cuyo tamaño es equivalente a una cantidad entera de celdas. El valor de los mensajes puede ser fijo o variable, dependiendo de los parámetros con los que se arrancó la simulación.

Para que las celdas contenidas en la cola local puedan ser transmitidas, es necesario que el terminal móvil envíe un SIG a la estación base. Una vez aceptado el SIG, la estación base enviará la señal de ACK al terminal móvil y simultáneamente tratará de asignar un canal para la transmisión de una celda. Apenas el terminal móvil consiga el permiso de transmisión, transmitirá una celda en el i-slot-up que le asignaron. Con la primera celda enviada viaja la información de cuantas otras celdas más se encuentran en espera de ser transmitidas (técnica de Piggybacking). La técnica de piggybacking tiene por objeto minimizar la cantidad de SIGs enviados a la estación base, aprovechando que ya se asignó un i-slot-up. Esta técnica tiene su aplicación durante la fase de reserva y también durante la fase de planificación, que será explicado mas adelante.

Al igual que en el caso de voz, si por alguna razón el terminal móvil no recibe la asignación de un i-slot-up antes del vencimiento de bwt, las celdas contenidas en la cola local comenzarán a ser eliminados por el terminal móvil, originándose pérdida de mensajes. El valor de bwt difiere dependiendo del tipo de tráfico generado: conversacional, interactivo, unidireccional, de fondo.

2.2.3.2.1 Fase de Reserva de Canal

Cada vez que el terminal móvil recibe celdas (de mensajes) provenientes del generador de tráfico, estas van a colocarse en la cola local por orden de llegada (FIFO). Seguidamente el terminal móvil cambiará del estado “libre” al estado “esperando”, con lo que se activa el FLAG que dará inicio al envío de una petición para transmisión. El diagrama de flujo de la Figura 2.16 esquematiza las acciones que suceden en el terminal móvil, cuando se inicia la generación de tráfico.

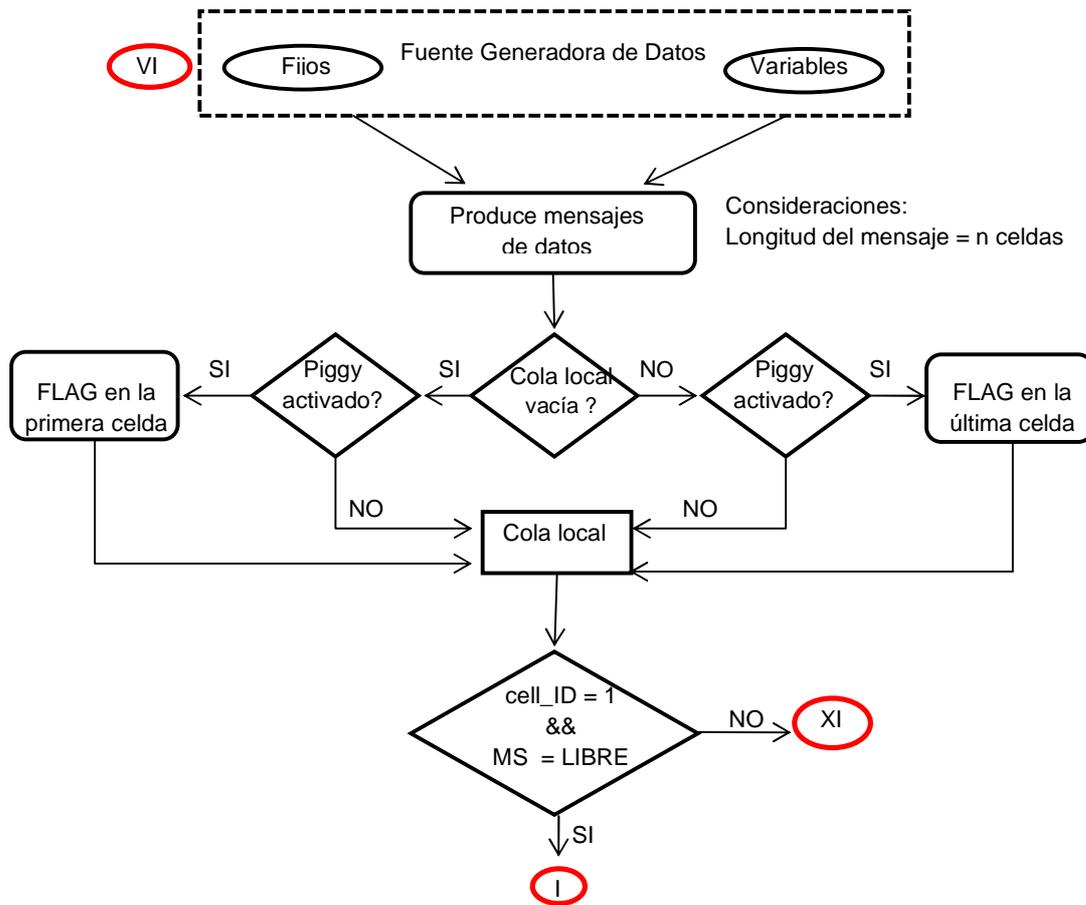


Figura 2.16 Fase de Reserva de Canal – Lado del Terminal Móvil (Funcionamiento para Datos, parte A)

Para realizar el envío de la petición, el terminal móvil debe esperar la llegada del R-slot. Una vez sintonizado el R-slot, el terminal móvil envía una petición llamada SIG en cuyo contenido se indica la dirección MAC del terminal móvil, el tipo de

fuentes de tráfico (voz, conversacional, interactivo, unidireccional, de fondo) y el código CDMA que le fue asignado al móvil durante la simulación.

Una vez enviado el SIG, se inicia el temporizador bwt cuyo valor se predefine en función del tipo de tráfico antes del arranque de la simulación. Simultáneamente, el terminal móvil empieza a escuchar el canal de bajada, esperando la llegada del A-slot con indicaciones provenientes de la estación base que le informarán si la petición SIG fue exitosa (ACK). La recepción de la señal ACK indicará al terminal móvil que la petición SIG fue recibida pero se encuentra en trámite para asignación del i-slot-up. Cuando recibe la señal de ACK, el terminal móvil inicia el monitoreo de todos los canales i-slot-down, con la finalidad de saber que recurso de i-slot-up le fue asignado. La estación base asignará el recurso de i-slot-up escribiendo implícitamente la ubicación en el i-slot-down

La explicación anterior es similar al funcionamiento para voz por lo que es válido usar el diagrama de flujo de la Figura 2.13 para resumir lo expuesto hasta aquí.

Al escuchar los slots de información pertenecientes al canal de bajada, el terminal móvil se enterará del i-slot-up asignado leyendo el campo "Dirección MAC de terminal exitoso" contenido en el i-slot-down. Todos los i-slot-down/i-slot-up están etiquetados con un identificador (id) único, por lo tanto el algoritmo de reserva lee el campo de id perteneciente al i-slot-down y aplica una lógica para asignar el i-slot-up (ver sección 2.2.1.1, Parámetros Generales, I-slot-offset).

Seguidamente el terminal móvil esperará la llegada del i-slot-up asignado para transmitir una celda de la cola local y de paso notificará a la estación base para que libere el recurso de i-slot-up y actualice la tabla de estados de i-slot-up. A diferencia del funcionamiento para voz, el terminal móvil no se queda con el uso exclusivo del i-slot-up asignado hasta vaciar su cola local, sino que las ranuras i-slot-up le son asignadas en cada inicio de trama. El diagrama de flujo de la figura 2.17 ilustra lo que se acaba de explicar.

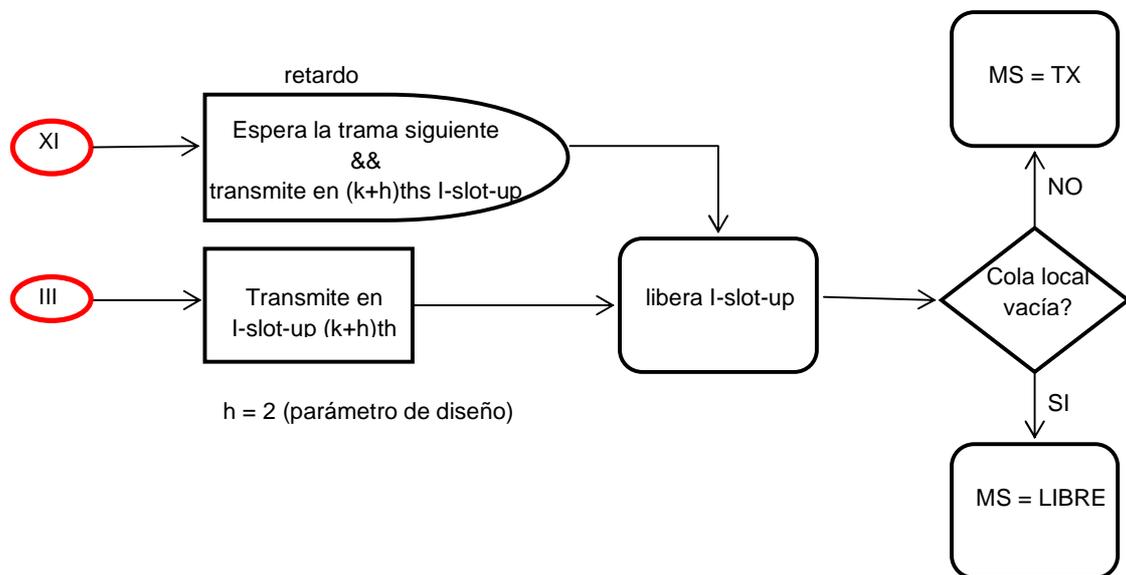


Figura 2.17 Fase de Reserva de Canal – Lado del Terminal Móvil (Funcionamiento para Datos, parte B)

Estrategia de Piggybacking

La estrategia de piggybacking hace uso de dos campos: el correspondiente al “bit de piggybacking” y el correspondiente al “contador de longitud de mensaje”. Ambos campos se ubican dentro de la cabecera de control del i-slot-up (sección 2.2.1.3). Dependiendo si la cola local se encuentra vacía o no, la estrategia de piggybacking funcionará de dos maneras.

Si la cola local no está vacía: cuando un mensaje de longitud n celdas ingresa a la cola local, el terminal móvil realizará la siguiente operación en el i-slot-up por donde se transmitirá la última celda del mensaje previo:

- activará el bit de piggybacking
- escribirá dentro del campo etiquetado como “contador de longitud de mensaje” la longitud del mensaje que acaba de ingresar.

Si la cola local se encuentra vacía: cuando un mensaje de longitud n celdas ingresa a la cola local, el terminal móvil realizará la siguiente operación en el i -slot-up por donde se transmitirá la primera celda del mensaje:

- activará el bit de piggybacking
- escribirá dentro del campo etiquetado como “contador de longitud de mensaje” la longitud del mensaje que acaba de ingresar menos una celda; es decir $(n-1)$.

La estrategia de piggybacking se esquematiza en la Figura 2.20.

2.2.3.2.2 Fase de Planificación

Para el caso de la estación base, la fase de planificación se inicia con la finalización del R-slot (inicio del A-slot). Durante la duración del R-slot, la BS recibe todas las señales SIG provenientes de todos los terminales móviles y los va almacenando en una cola temporal, al término del R-slot realiza una clasificación de los SIGs recibidos en función del tipo de tráfico peticionado: paquetes o mensajes. Paralelamente envía la notificación de ACK usando el A-slot del canal de bajada para avisar a las terminales móviles que el pedido de SIG se encuentra en trámite. Seguidamente coloca todos los SIGs cuya petición es para paquetes, en una cola FCFS (first coming, first served) mientras que los SIGs cuya petición es para mensajes son colocados en cuatro colas, para ser atendidos en el siguiente orden: primero la cola conversacional, segundo la cola interactiva, tercero la cola unidireccional y cuarto la cola de fondo. Al interior de cada cola mencionada, la atención de peticiones será realizada en forma circular RR (*round robin*). El diagrama de la Figura 2.18 esquematiza lo mencionado anteriormente, donde primero se atiende el tráfico de voz para luego atender el tráfico de datos, que es atendido tomando en cuenta un orden de prioridad.

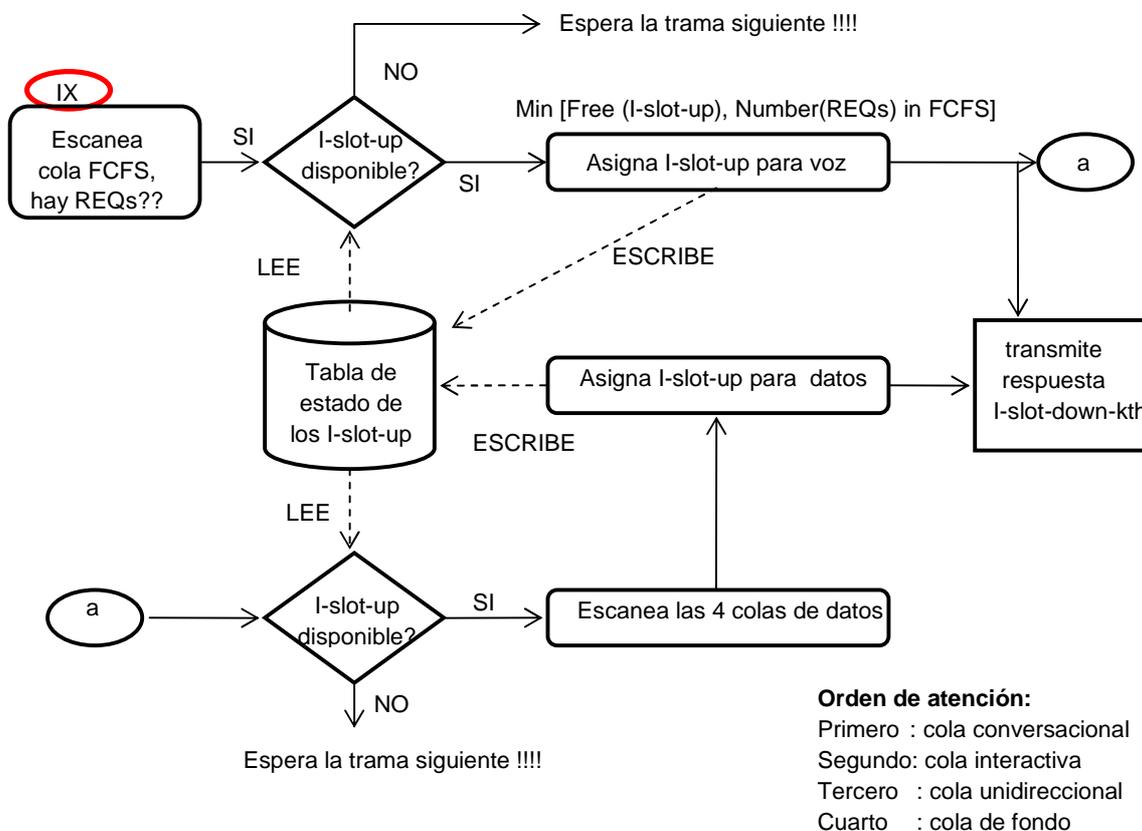


Figura 2.18 Fase de Planificación – Lado de la Estación Base (Funcionamiento para Datos – parte A)

Estrategia de Calidad de Servicio (QoS): Clasificación de peticiones SIG

Por cada SIG de datos recibido, la estación base primero identificará el tipo de tráfico (si es conversacional, interactivo, unidireccional o de fondo), luego creará una cola tipo FCFS, que identificará al terminal móvil solicitante, y colocará un elemento, llamado REQ, dentro de la cola creada. Luego la cola FCFS se insertará en una de las cuatro colas de calidad de servicio dependiendo del tipo de tráfico identificado en el SIG.

Es así que al final del análisis de los SIGs de datos, en la estación base habrán por cada clase de servicio, una cierta cantidad de colas FCFS; cada una de estas tendrá una cierta cantidad de REQs, tal como se muestra en la figura 2.19.

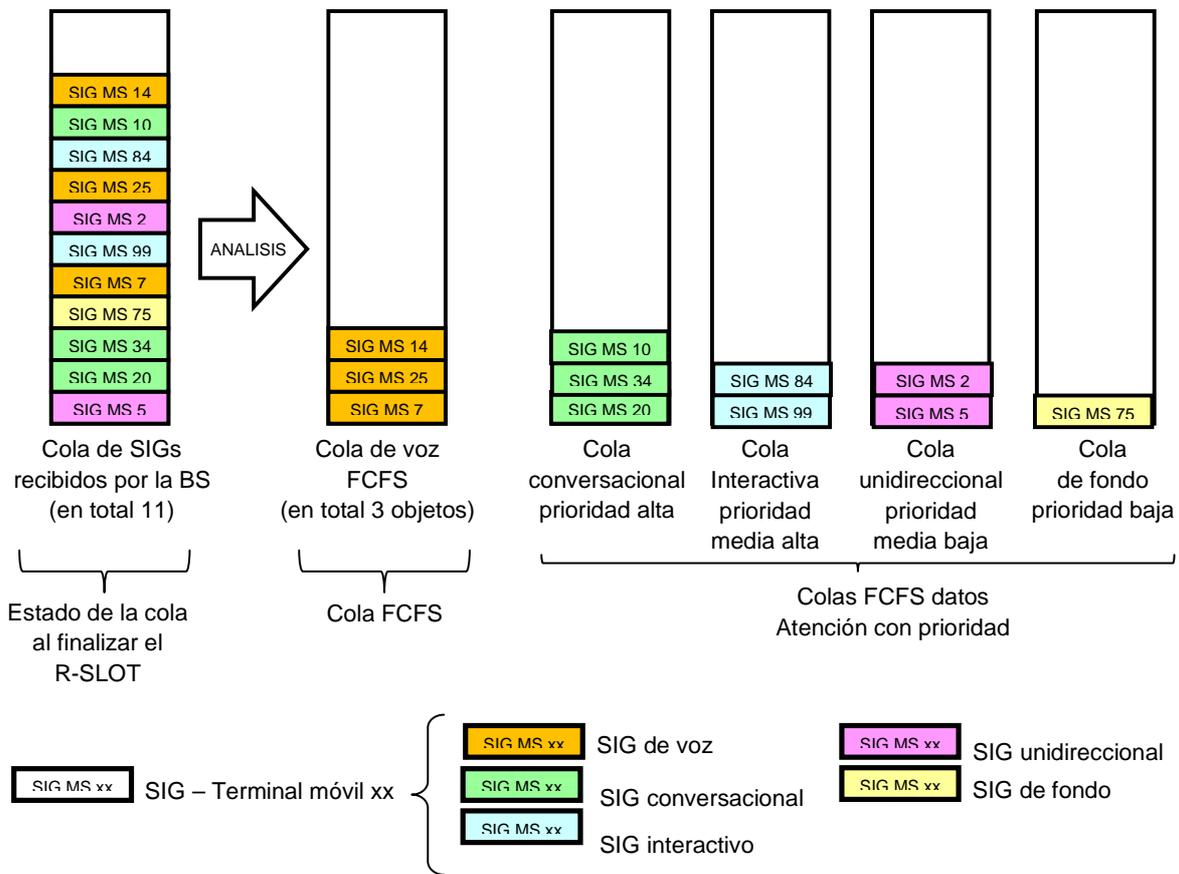


Figura 2.19 Fase de Planificación – Lado de la Estación Base: Estrategia QoS, Clasificación de SIGs

Estrategia de Calidad de Servicio (QoS): priorización en la asignación de recursos

Seguidamente la BS consultará la base de datos que contiene el estado de los i-slot-up (estados permitidos: libre y ocupado) y asignará un i-slot-up siguiendo el orden indicado:

- Primero, se atenderá la cola FCFS de voz y asignará los i-slots-up que pueda encontrar libres
- Segundo, dentro de la cola conversacional, asignará los i-slot-up a cada cola FCFS de datos pero siguiendo el orden circular (round robin); cuando todas las colas FCFS hayan sido servidas pasará al siguiente paso.

- Tercero, dentro de la cola interactiva, asignará los i-slot-up a cada cola FCFS de datos pero siguiendo el orden circular (round robin); cuando todas las colas FCFS hayan sido servidas pasará al siguiente paso.
- Cuarto, dentro de la cola unidireccional, asignará los i-slot-up a cada cola FCFS de datos pero siguiendo el orden circular (round robin); cuando todas las colas FCFS hayan sido servidas pasará al siguiente paso.
- Quinto, dentro de la cola de fondo, asignará los i-slot-up a cada cola FCFS de datos pero siguiendo el orden circular (round robin).

En caso que la BS no encuentre recurso disponible (ausencia de i-slot-up en estado LIBRE) entonces deberá esperar la próxima trama para realizar nuevamente la asignación. Si la BS encuentra un i-slot-up disponible entonces asigna un i-slot-up escribiendo la dirección MAC del terminal exitoso en el campo correspondiente del i-slot-down.

Cabe resaltar que la BS va ir actualizando la tabla de estado de i-slot-up cada vez que recibe tráfico proveniente de los terminales móviles (MS). Justamente la instancia del protocolo MAC eficiente que reside en el MS, está preparada para enviar una notificación de liberación de i-slot-up cuando ya no tiene más paquetes por enviar. Al recibir la señal de liberación, la estación base procederá a eliminar la cola FCFS de datos que fue creada para el terminal móvil en particular.

Estrategia de Piggybacking

El esquema de la Figura 2.19 se presenta para el caso en que la estación base recibe los SIGs provenientes de los terminal móviles. En esta situación el algoritmo de planificación va creando colas FCFS para cada terminal móvil e insertando un objeto REQ dentro de cada cola FCFS mencionada.

Existe otra situación en que la estación base insertará objetos REQs dentro de las colas FCFS de datos que ya se encuentran creadas, y justamente está relacionada a la estrategia de piggybacking.

Con el piggybacking se notificará a la estación base de la presencia de celdas adicionales en la cola local del terminal móvil. Dicho aviso aparecerá con la llegada

de la última celda del mensaje que fue transmitida por el terminal móvil hacia la estación base. Es así que la última celda (contenida en el i-slot-up) informará sobre la existencia de mensajes adicionales pendientes de envío y la estación base se encargará de llenar las colas FCFS de datos correspondientes a cada móvil con la cantidad de REQs indicados en el campo “contador de longitud de mensaje”. Finalmente una cola FCFS de datos puede ser alimentada por un REQ (al final de un R-slot) o por uno o más REQs al final de un i-slot-up, a través de la estrategia de piggybacking.

Esta situación se esquematiza en la figura 2.20 y se refuerza con el diagrama de flujo de la figura 2.21.

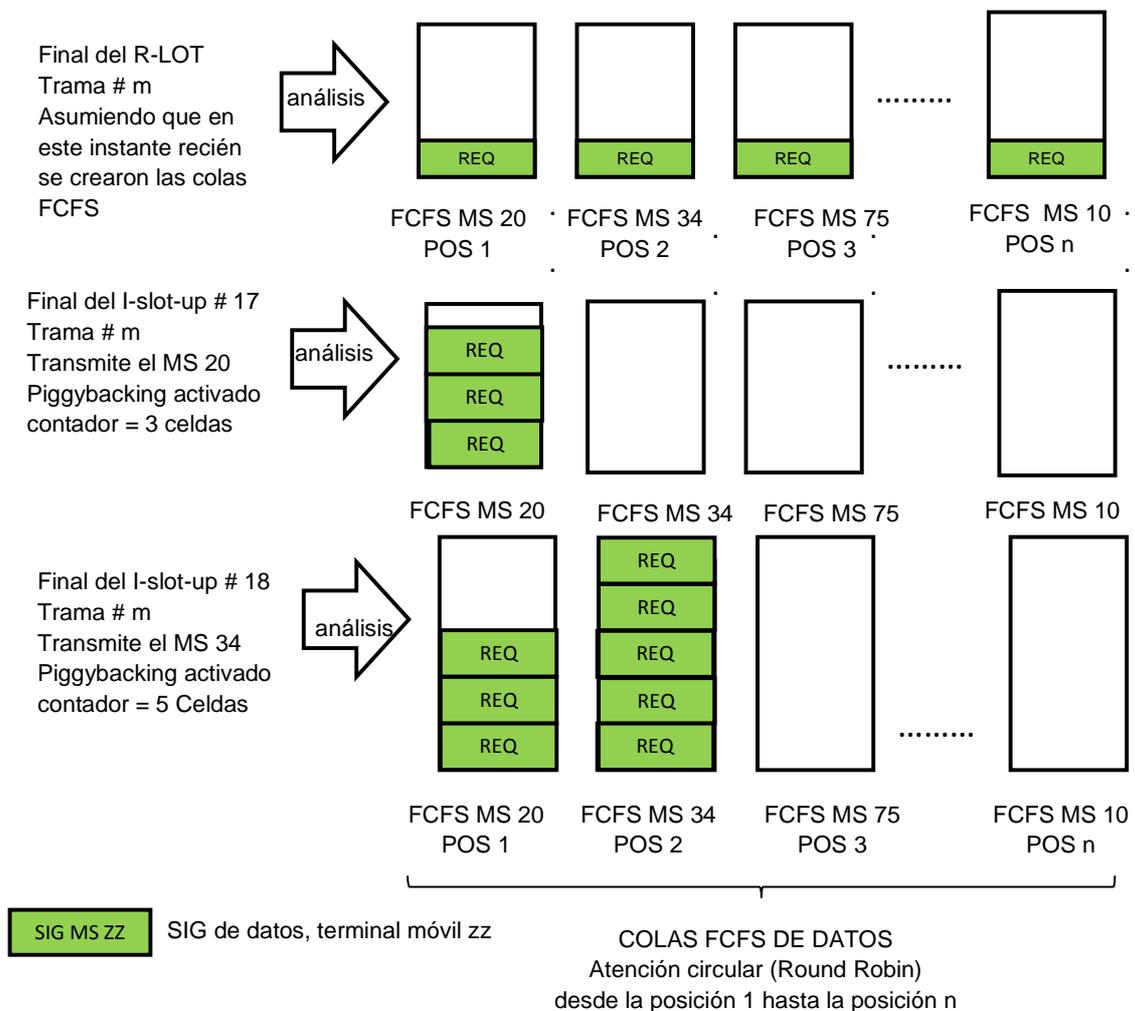


Figura 2.20 Fase de Planificación – Lado de la Estación Base: Estrategia de piggybacking

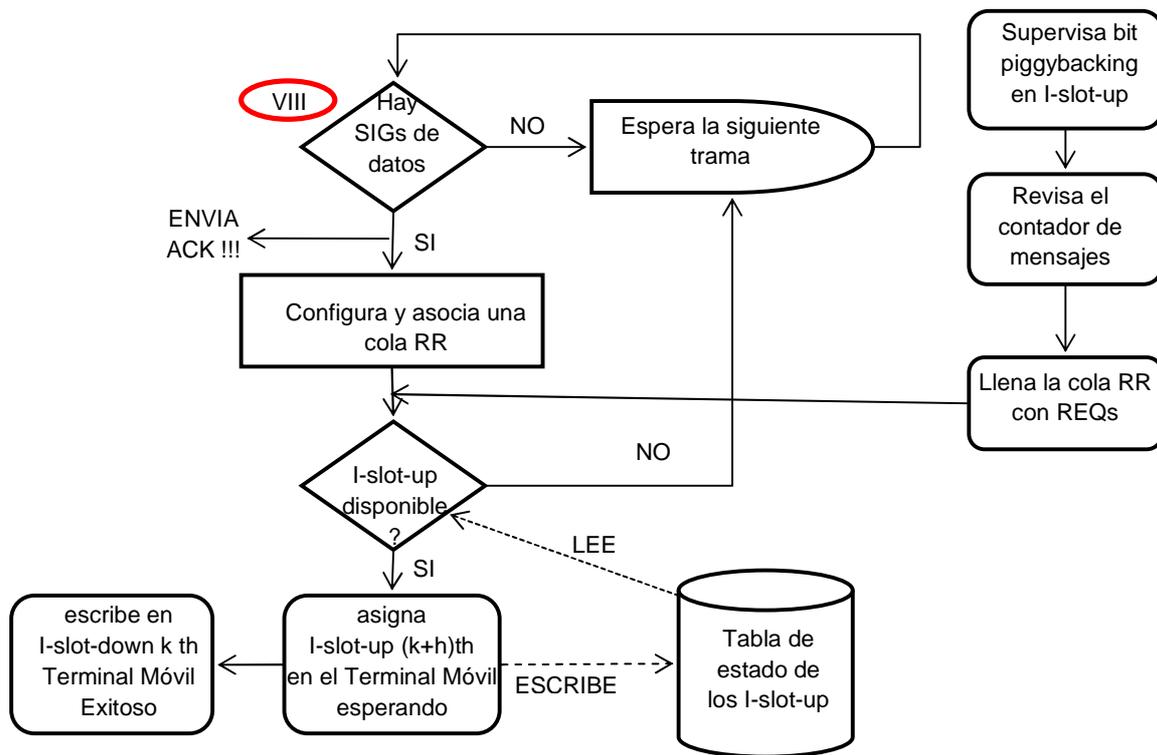


Figura 2.21 Fase de Programación – Lado de la Estación Base:
(Funcionamiento para Datos – parte B)

2.2.4 Diagramas de Señalización, y Temporizadores

En la sección 2.2.3 se detalló el funcionamiento del protocolo MAC eficiente, apoyados en los diagramas de flujo, que facilitaron su comprensión. Parte de los modelos lógicos que ayudaron a entender el funcionamiento de [2] y que sirven para exponer el funcionamiento de nuestro protocolo, lo constituyen los diagramas de señalización y de temporizadores.

Diagramas de Señalización

El diagrama de señalización constituye una herramienta valiosa para la comprensión de la lógica de algunos procesos; como por ejemplo, el funcionamiento de un protocolo. El diagrama de señalización ayuda a entender el proceso de comunicación entre entidades, aportando una idea clara de los estadios, mensajes y eventos que ocurren durante la interacción de las partes

involucradas. Para el caso de esta tesis, el diagrama de señalización expone las actividades del protocolo, que ocurren durante la comunicación entre el terminal móvil y la estación base.

Para el caso de voz, el diagrama de señalización de la figura 2.22, muestra desde el inicio hasta el fin, las acciones que se suscitan entre el terminal móvil (MS) y la estación base (BS) apenas el generador de tráfico, en el MS, inicia la producción de paquetes (estado ON), hasta vaciar la cola local (estado OFF).

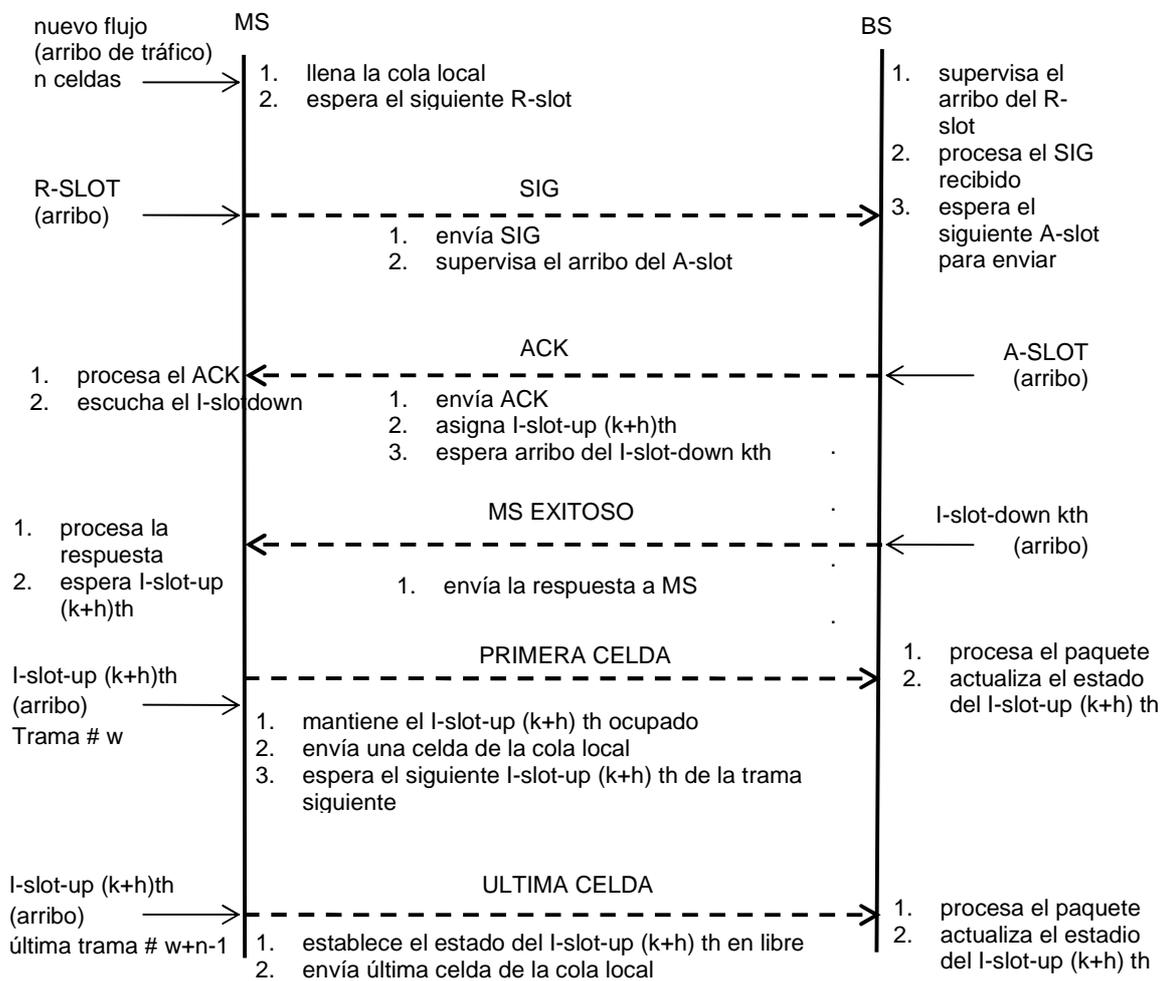


Figura 2.22 Señalización entre MS y BS – Caso Voz

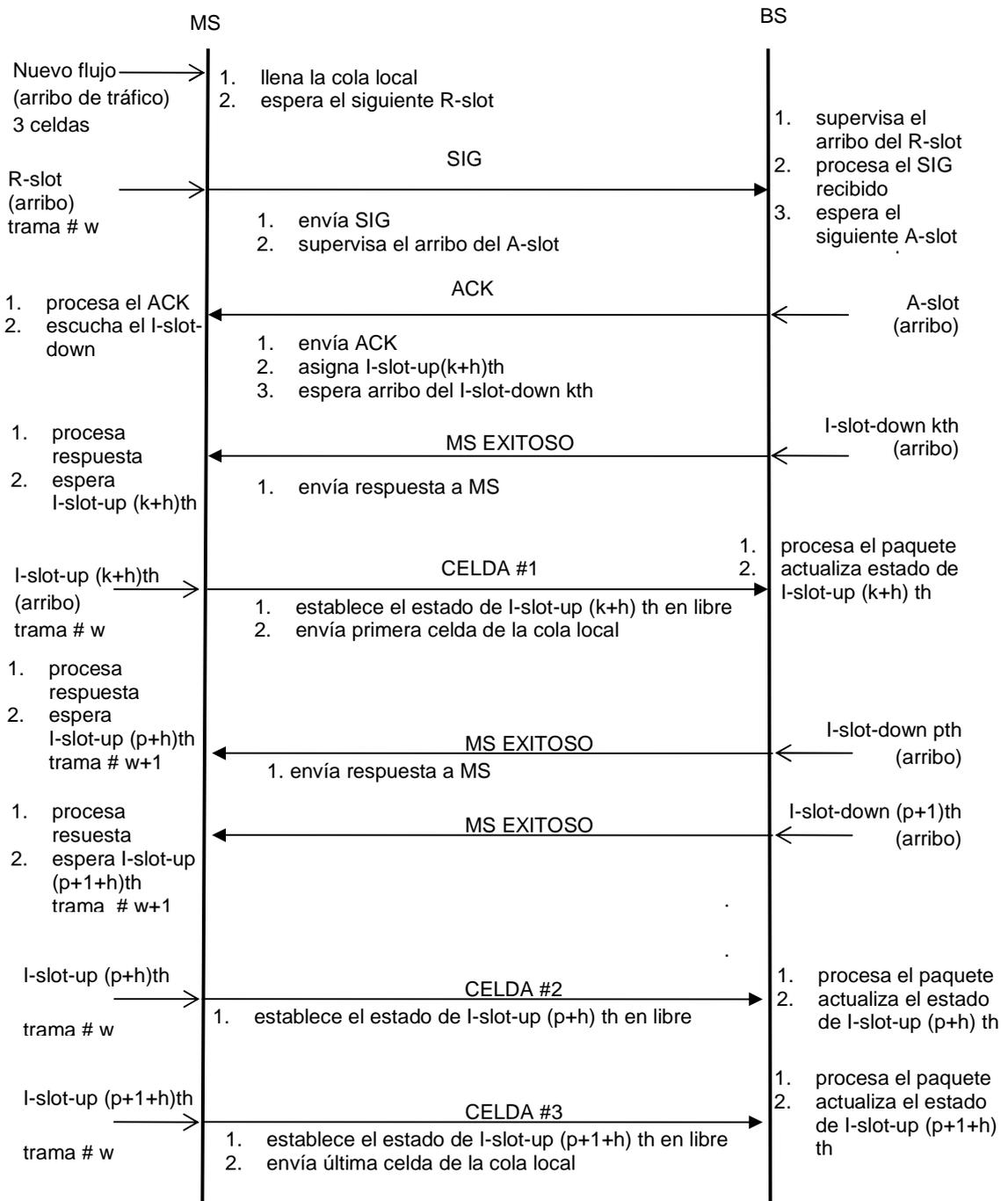


Figura 2.23 Señalización entre MS y BS – Caso Datos de Velocidad Binaria Variable

Para el caso datos con velocidad binaria variable, la figura 2.23 muestra el diagrama de señalización en el que se detallan las acciones y reacciones entre el terminal móvil y la estación base. En este diagrama el terminal móvil genera 3 celdas, la primera se entrega en la trama w y la segunda y tercera se entregan en la trama $w+1$.

Temporizador

Como se indicó en la sección 2.2.1.1, dentro de Parámetros Generales, se define un temporizador llamado bwt , con la finalidad de asegurar que los paquetes y mensajes sean enviados dentro de los parámetros de tolerancia permitidos. En ese sentido el valor de bwt dependerá del tipo de tráfico que se esté generando. El temporizador bwt cumple la función de eliminar aquellas celdas que hayan excedido el tiempo máximo de permanencia dentro de la cola local. Esta situación puede llegar a darse por dos motivos:

- Primer caso, ocurrido por falla en la recepción del ACK o por retardo en la respuesta. Esto se debe a fallos en el canal de comunicación que pueden provocar que la solicitud de R-slot no sea recibida por la estación base o viceversa, es decir que provoquen que el A-slot enviado nunca sea recibido por el terminal móvil. Para el caso particular de la figura 2.24 mostrada, el valor de bwt es equivalente a la duración de dos tramas de subida.
- Segundo caso, ocurrido por retardo en la respuesta de la estación base (BS) para la asignación del I-slot-up. Esto se debe a que la estación base no puede encontrar recursos de i-slot-up disponibles para asignarlos al terminal móvil. El retardo en la asignación de slots da lugar a que el temporizador empiece a eliminar los mensajes de la cola local que han excedido el valor máximo permitido en tiempo. Ver la figura 2.25.

2.2.5 Métricas usadas

Para medir el desempeño del protocolo MAC eficiente para redes inalámbricas es importante separar los medidores estadísticos en función del tráfico que se transmite. Es así que se tienen métricas del desempeño: para paquetes y mensajes; debido a que tendrán un tratamiento distinto por parte del protocolo.

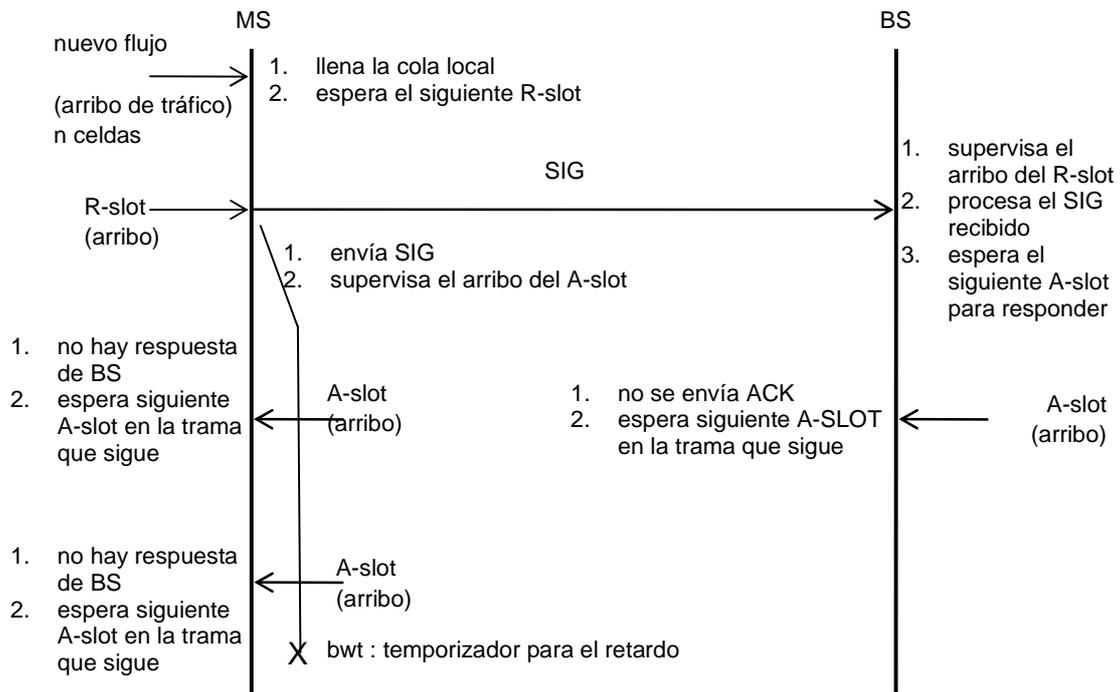


Figura 2.24 Temporizador – Primer Caso

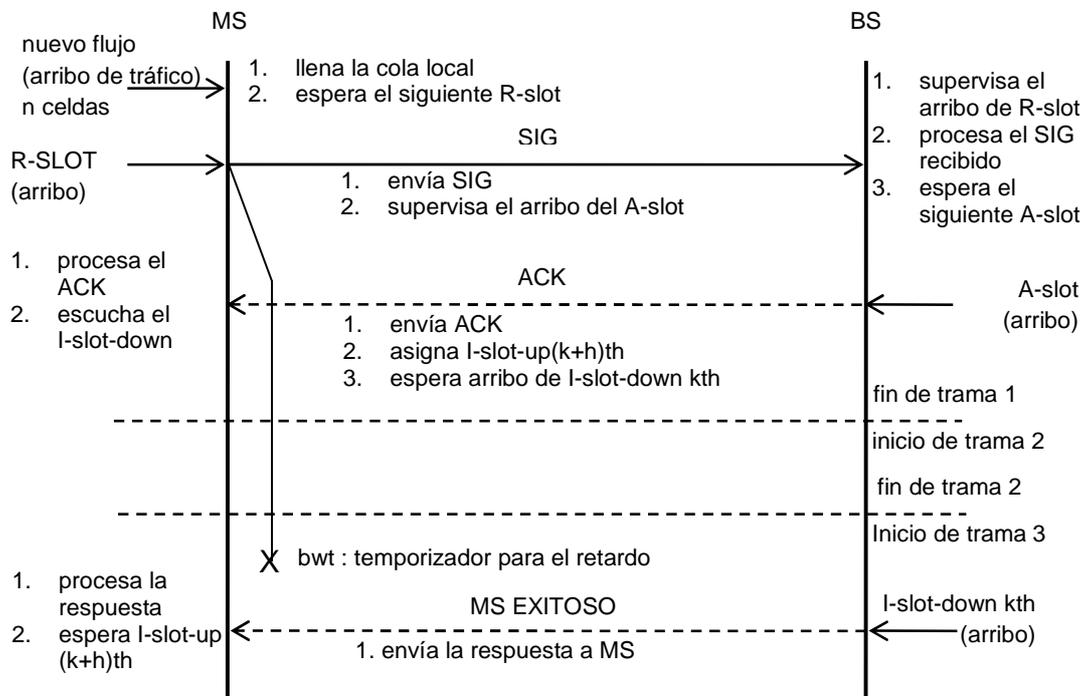


Figura 2.25 Temporizador – Segundo Caso

Métricas para voz

- Pérdida de los paquetes de voz (en inglés packet dropping), mide la fracción de paquetes cuyo retardo superó el valor umbral bwt y consecuentemente fueron eliminados.
- Tiempo promedio del retardo de voz, mide el tiempo promedio entre el instante en que se creó el paquete y el instante que fue recibido por la estación base.
- Desempeño (Throughput) Promedio, número de bits exitosamente transmitidos por todos los terminales móviles y por unidad de tiempo.

Métricas para datos

- Promedio y Distribución del Retardo de Acceso, tiempo necesario para la transmisión completa del mensaje, desde el instante en que se inserta en la cola local del terminal móvil hasta que es recibido por la estación base..
- Desempeño (Throughput) Promedio, número de bits exitosamente transmitidos por todos los terminales móviles y por unidad de tiempo.
- Factor de Utilización, fracción de i-slot-up utilizado durante la simulación.

2.3 Implementación del protocolo MAC en OMNET++

El protocolo MAC desarrollado en esta tesis, se confinó en un ambiente de simulación, para esto se usó el Framework de OMNET++ [3], que es un simulador de eventos discretos. Para detalles acerca de OMNET++ vea el Anexo D.

El protocolo MAC fue desarrollado en lenguaje C++, y su realización fue posible gracias al diseño que se elaboró y que se ha explicado en la sección 2.2. Los diagramas de flujo, diagramas de señalización, diagramas de funcionamiento de temporizadores, fórmulas para diseñar la trama y el detalle de la estructura interna de las ranuras; todo lo mencionado contribuyó a programar la lógica de funcionamiento del protocolo. También los modelos de fuentes de tráfico fueron programados usando la lógica de estados finitos y que emula el funcionamiento de un generador de tráfico con estados de encendido y apagado (ON/OFF).

2.3.1 Ambiente de Simulación

El ambiente de simulación está conformado por todos los componentes necesarios que habilitan el escenario de la problemática y constituyen la topología de la red ad-hoc. La topología cobra vida a través de dos elementos: la trama TDMA que se regenera cada intervalo de tiempo y las fuentes de tráfico que generan eventos de forma aleatoria.

Esquema de simulación propuesto

La topología de nuestra red ad-hoc está conformada por:

- Una instancia que simula la Estación Base (módulo compuesto)
- Varias instancias que simulan los Terminales Móviles (un módulo compuesto por cada instancia)
- La simulación de la Trama CRDA (algoritmo)
- Varias instancias que simulan las fuentes de tráfico (un módulo simple por cada instancia)

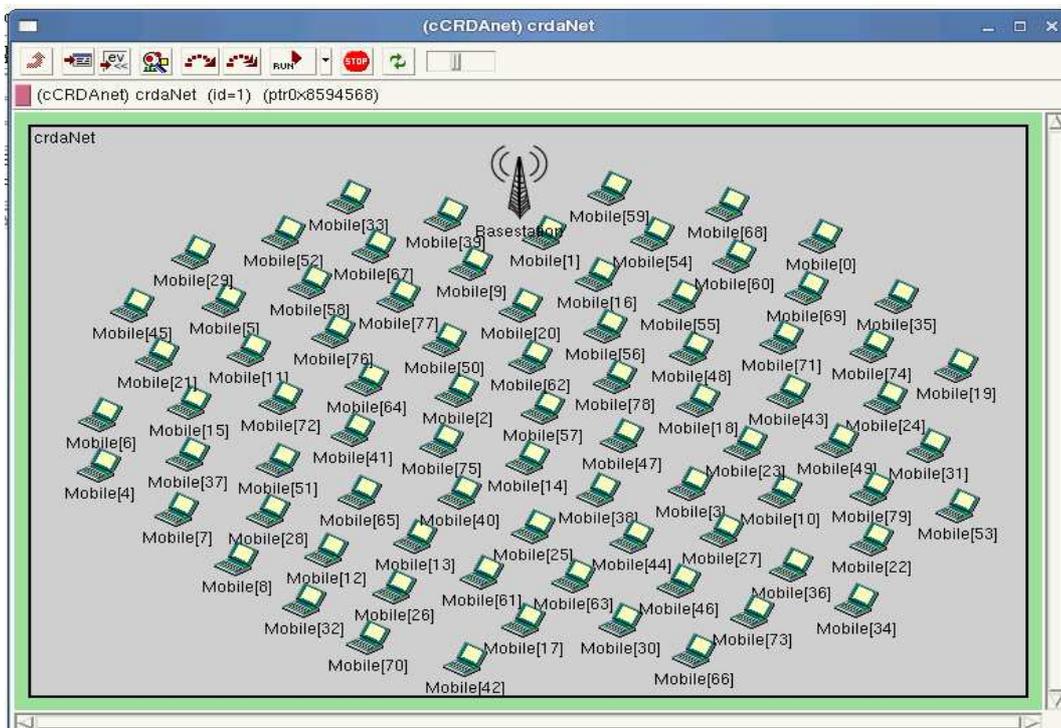


Figura 2.26 Topología en OMNET++

La figura 2.26 muestra como luce el diagrama topológico en OMNET++. Se observa que existe un módulo de estación base y varios módulos de terminales móviles. El número de terminales móviles que forman parte de la topología se controla modificando el archivo de parámetros omnetpp.ini (sección 2.1.4.1). Los simuladores de trama no se muestran debido a que son algoritmos que funcionan dentro de los módulos de estación base y de terminal móvil. El generador de tráfico, es un módulo simple e interno al módulo de terminal móvil.

Archivos del Protocolo

Como se indicó en la sección 2.1.4.1, la simulación en OMNET++ requiere la existencia de ciertos archivos fuente. Para el caso del protocolo MAC eficiente, de esta tesis, los archivos que presentamos a continuación proveen el funcionamiento del protocolo y son fundamentales para generar el archivo ejecutable donde fueron probadas las simulaciones.

- **Archivo phy.cc**, constituye un módulo simple dentro de OMNET++ y define el comportamiento de la capa física. Aquí se ha programado un algoritmo que simula las tramas CRDA tanto del enlace de subida (uplink) como la del enlace de bajada (downlink). El algoritmo de la trama también simula las ranuras R-slot, A-slot, I-slot-up, I-slot-down, con las estructuras internas definidas en la sección 2.2.1.3. Este módulo es polimórfico en el sentido que puede tomar una configuración diferente cuando se inicia como una instancia de estación base o una de terminal móvil. Este archivo define la clase cPhy.
- **Archivo mac.cc**, constituye un módulo simple dentro de OMNET++ y define el comportamiento de la capa MAC; en otras palabras aquí radican los algoritmos que hacen funcionar al protocolo. En este archivo se han programado los algoritmos de reserva y planificación; el primero encargado de solicitar i-slot-up para transmitir, y el segundo para asignarlos. Este módulo también es polimórfico y puede tomar un comportamiento distinto cuando se inicia como una instancia de estación base o de terminal móvil. Este archivo define la clase cMac.

- **Archivo traffic.cc**, constituye un módulo simple dentro de OMNET++ y define el comportamiento de la capa de tráfico. Aquí se han programado los algoritmos de las fuentes de tráfico para voz y datos con velocidad variable. Este módulo también es polimórfico, cuando se levanta como una instancia del terminal móvil funciona como generador de tráfico y cuando se inicia como una instancia de estación base funciona como sumidero. Este archivo define la clase cTraffic.
- **Archivo network.ned**, define la topología de la red ad-hoc simulada. Dentro de este archivo se crean los enlaces entre las 3 clases cPhy, cMac y cTraffic. Aquí también se declaran las instancias de terminal móvil y de estación base; ambas instancias son módulos compuestos cuyos módulos simples son las 3 clases que definen la capa física, capa mac y capa de tráfico.
- **Archivo omnetpp.ini**, en este archivo se ingresan los parámetros de la simulación como: cantidad de terminales móviles, tipos de tráfico (voz, conversacional, interactivo, unidireccional o de fondo), activación del piggybacking, tamaño de los mensajes de datos (expresado en celdas de 68 bits), valor del temporizador bwt, tiempo de la simulación, entorno de simulación (gráfico o consola), activación de los colectores estadísticos y los iconos en el modo gráfico.
- **Archivo messages.msg**, aquí se definen estructuras de los mensajes (eventos) que se pasarán entre los módulos. Estos mensajes vienen a ser derivados de la clase básica cMessage que provee OMNET++. En este archivo se programa la estructura interna de la trama y de cada ranura del protocolo MAC eficiente.
- **Archivo mobile.txt**, es un archivo de texto que contiene parámetros útiles para la simulación de los terminales de voz. Estos parámetros se cargan al inicio de la simulación. Los parámetros que almacena son: código CDMA del terminal móvil, dirección MAC del terminal móvil, identificador (id) del terminal móvil.

Integración del Protocolo con el Ambiente de Simulación

Los módulos simples que fueron invocados en el archivo network.ned, enlazan las clases cPhy, cMac y cTraffic con el Framework del simulador. Las clases cPhy, cMac y cTraffic fueron programadas como clases derivadas de la clase básica cSimpleModule (del framework de OMNET++). Cuando se programa un archivo con extensión *.cc dentro de OMNET++, es importante declarar las librerías del simulador, que están contenidas en la cabecera omnetpp.h.

La integración de los archivos del protocolo MAC con el framework de OMNET++ se muestra en la figura 2.27. Los archivos *.cc contienen las declaraciones y definiciones de las clases que derivan de cSimpleModule. Note que las clases cBasestation y cMobile se crean a partir de tres clases derivadas: cPhy, cMac y cTraffic. Las clases cBasestation y cMobile hacen referencia a módulos compuestos y se crean indirectamente en el archivo network.ned. Por otro lado las clases cBasestation y cMobile se integran a la clase cECRDAnet, esta última clase define la topología de la red ad-hoc y también se crea indirectamente en el archivo network.ned. Finalmente todas las clases se integran al framework de OMNET++..

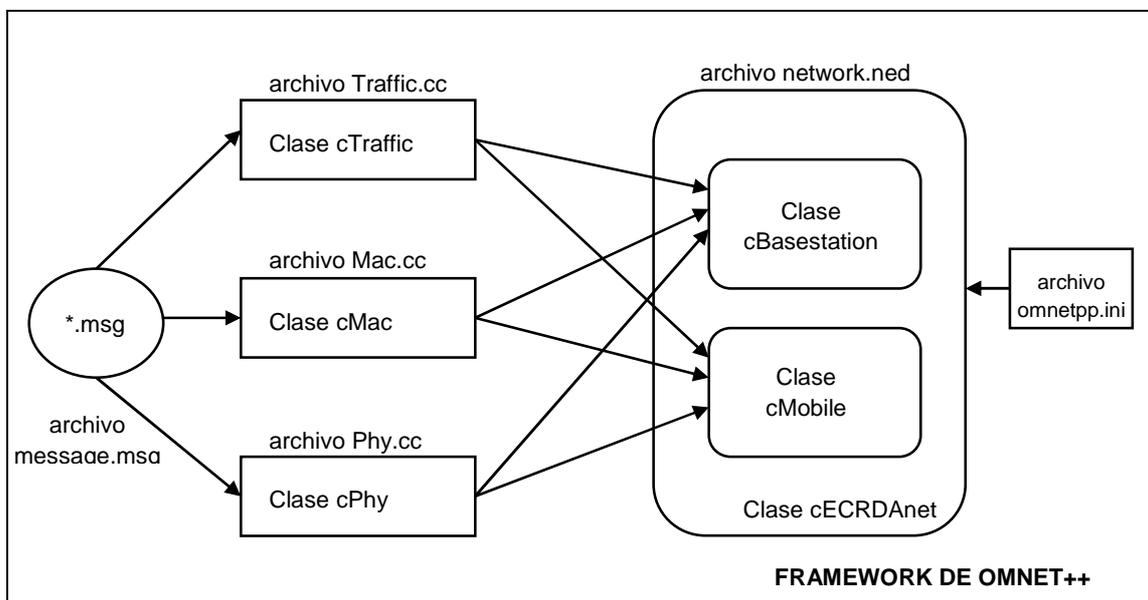


Figura 2.27 Integración del Protocolo con el framework de OMNET++

2.3.2 Módulos del Protocolo

En esta sección se ahonda en los detalles de las instancias que conforman la topología de la red ad-hoc simulada (sección 3.1, Ambiente de Simulación – Esquema de simulación propuesto). Dichas instancias representan objetos (clases) en el framework de simulación y son visibles al usuario (Figura 2.26). Los objetos programados fueron dos: el objeto estación base (clase cBasestation) y el objeto terminal móvil (clase cMobile). Estos objetos se programaron como módulos compuestos de OMNET++ y son derivados de la clase cCompoundModule. También es posible inspeccionar visualmente el contenido de los objetos estación base y terminal móvil, tal como lo muestra la figura 2.29. Para comunicar los módulos estación base y terminal móvil ha sido necesario programar la trama CRDA con todas sus estructuras internas, para ello se re-definió la clase cMessage, adaptándola a las características de la trama CRDA.

Para llevar a cabo la implementación del programa que simula el protocolo MAC eficiente usando OMNET++, se plantearon los siguientes requerimientos de diseño

- **Modularidad de los componentes:** se requería que los objetos estación base y terminal móvil sean modulares. De esta manera tomando como inspiración el concepto de capas en telecomunicaciones (modelo OSI), se programó una capa física, una capa mac y una capa generadora de tráfico. Todas las capas se enlazan jerárquicamente para formar: ó un objeto estación base ó un objeto terminal móvil.
- **Polimorfismo de los componentes:** Los objetos estación base y terminal móvil deberían derivar de la misma instancia base. Para nuestro caso, el “módulo compuesto” toma dos comportamientos distintos, de acuerdo a la función que realiza; es decir si es terminal móvil o estación base.

La figura 2.28 ilustra la propiedad de modularidad del protocolo MAC, mientras que en la figura 2.29 se puede observar como lucen los módulos, de estación base y de terminal móvil, cuando ya han sido programados en OMNET++. La propiedad de polimorfismo se ilustra en la figura 2.27, donde las clases cBasestation y cMobile hacen uso de las mismas clases base (cPhy, cMac, cTraffic).

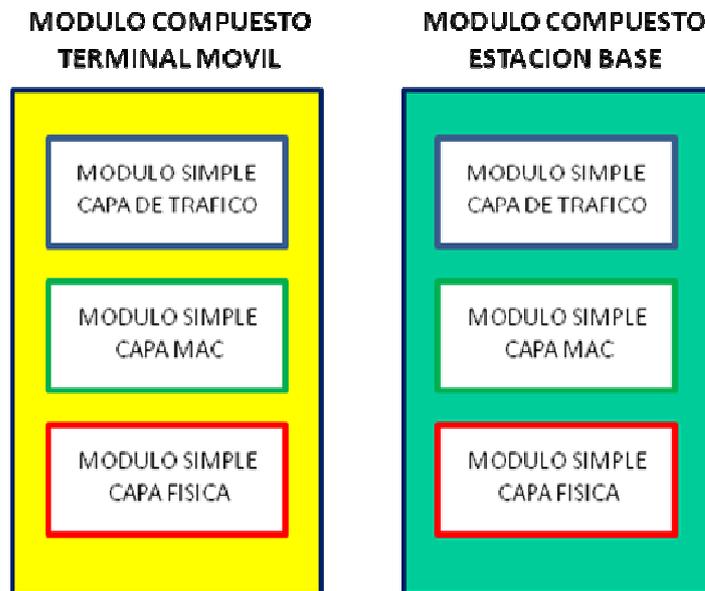


Figura 2.28 Módulos del Protocolo – Módulos Simples y Módulos Compuestos

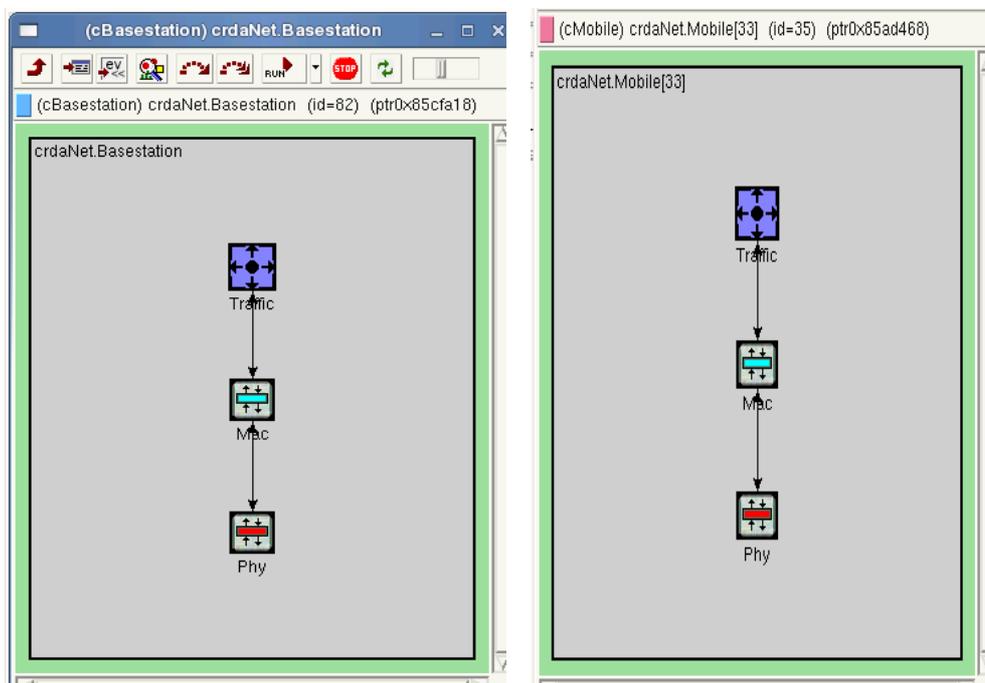


Figura 2.29 Módulos Estación Base (Izq.) y Terminal Móvil (Der.) en OMNET++

Las clases cBasestation y cMobile, por ser módulos compuestos, encapsulan y enlazan a los módulos simples que forman parte de estos. De esta manera la propiedad de polimorfismo está en cada módulo simple que forma parte del módulo compuesto. Como se ha mencionado, en los módulos simples se programan los algoritmos que rigen las funciones y propósitos de los módulos.

A continuación se detallan los elementos fundamentales que forman parte del programa que simula el protocolo MAC de esta tesis:

- Mensajes: redefinición de la clase cMessage
- Módulo simple de Capa Física: clase cPhy
- Módulo simple de Capa Mac: clase cMac
- Módulo simple de Capa de Tráfico: clase cTraffic

Definición de Mensajes

Clase : CMessage (redefinido)

En OMNET++, los eventos discretos ocurren por la llegada de mensajes (msg) a los módulos que se han definido en la topología. Dichos mensajes contienen campos por defecto que son usados por el programador de acuerdo a su conveniencia. En nuestro caso ha sido necesario re definir los mensajes a fin de agregarle campos adicionales que nos ayudan a lograr la simulación que esperamos. Se definieron tres tipos de mensajes: mensajes de control, mensajes de protocolo y mensajes de tráfico. La figura 2.30 ilustra el contenido de un mensaje personalizado que se ha utilizado (para este ejemplo se trata de un mensaje de control). Se observa como es posible inspeccionar los mensajes a un nivel detallado.

control

Son mensajes de control y se usan para generar las tramas de subida y de bajada (tramas CRDA), proveen el sincronismo al terminal móvil y estación base. Los mensajes de control se generan internamente en la capa física y envían los

eventos a la capa MAC para que esté al tanto de la llegada de los slots (R-slot, A-slot, I-slot-up, I-slot-down).

Campos:

- int tipo enum(tipo_slot); de tipo entero y se usa para especificar si se genera un slot de tipo R-, A- ó I-
- int disparo enum(intervalo); de tipo entero y se usa para delimitar el tiempo de duración de un slot.
- int i_slot_id; de tipo entero, se usa para etiquetar y enumerar los slots de información (up o down).
- int tipoframe enum(tipo_frame); de tipo entero, se usa para especificar el tipo de trama: de subida o de bajada (uplink o downlink)
- bool fin_frame; de tipo lógico, se usa para indicar el final de una trama

protocol

Es un mensaje de protocolo que conforma la estructura interna de las ranuras (sección 2.2.1.3), es decir lleva los campos que conforman la ranura tipo R-slot, A-slot ó I-slot-up, I-slot-down. Cuando se trata de una ranura R-slot, el mensaje toma los valores para este tipo de ranura en particular; haciendo lo mismo para las ranuras A-slot, I-slot-up ó I-slot-down. Es un mensaje polimórfico.

En el mensaje de tipo protocol se ha obviado el campo de “carga útil” ya que no es necesario para el objetivo de la tesis

Campos:

- int tipomac enum(tipo_mac); de tipo entero y especifica si el mensaje enviado es un SIG, un ACK, un PACKET (paquete de voz), un MESSAGE (mensaje de datos) ó se trata de un ISLOT (slot de información)
- int tosmac enum(tos_mac); de tipo entero y especifica si el tipo de servicio TOS, ayuda a diferenciar si el terminal móvil es de voz o de datos.
- int ms_address_tagged; de tipo entero y guarda la dirección MAC del terminal móvil

- int ms_codebook; de tipo entero y guarda el código CDMA del terminal móvil
- int ms_id; de tipo entero y guarda el identificador numérico (id) del terminal móvil
- int ms_moduleid; de tipo entero y es usado internamente en la simulación para identificar un módulo.
- int ms_islotup; de tipo entero y es usado para identificar si el slot de información es de subida o de bajada.
- bool ms_status_islotup; de tipo lógico y es usado para notificar a la estación base sobre el estado del slot de información de subida.
- long ms_msglength; de tipo long double y es usado para informar a la estación base el tamaño del mensaje de datos.
- int ms_msgorder enum(msg_order); de tipo entero y es usado internamente en la simulación para notificar el orden de la celda dentro de la cola local en el terminal móvil.
- simtime_t ms_creation_time; de tipo simtime (tipo definido de OMNET++) y es usado para especificar el instante de creación de una celda.
- bool ms_piggybacking; de tipo lógico y es usado para notificar sobre el encendido de la funcionalidad de piggybacking.
- int ms_piggylength; de tipo entero, usado para especificar la longitud de del mensaje cuando se habilita el piggybacking
- bool ms_dummy; de tipo lógico, es usado internamente en la simulación para tratar condiciones excepcionales.
- bool ms_acksent; de tipo lógico, usado para actualizar el estado del terminal móvil cuando recibe la señal de ACK. Es interno a la simulación.
- simtime_t ms_queuing_time; de tipo simtime y se usa para guardar el registro del tiempo de encolamiento para las celdas contenidas en la cola local del terminal móvil

traffic

Es un mensaje de tráfico usado por el módulo de tráfico para informar a la capa MAC acerca del estado del generador de tráfico, informándole si se encuentra encendido o apagado. Es un mensaje interno a la simulación.

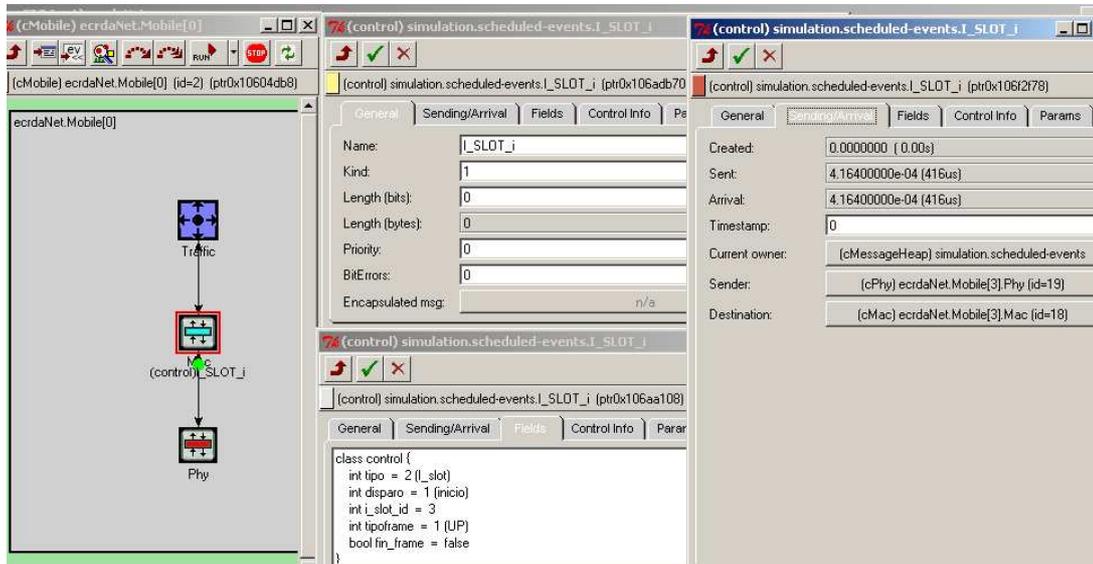


Figura 2.30 Mensajes en OMNET++

Módulo Simple Capa Física

Clase : cPhy

Es una clase derivada de cSimpleModule

Define el comportamiento de la capa física para la estación base y el terminal móvil. Esta capa se encarga de generar la trama TDMA cada cierto intervalo de tiempo y comunica ello a la capa mac.

Variables Generales

- struct MobProp; estructura de almacenamiento temporal de la data de Mobiles: dirección MAC, código CDMA e identificador id.
- const char *mName; variable tipo caracter que clasifica el tipo de modulo (BS o MS)
- int mld; variable tipo entera que almacena el ID del módulo MS
- bool isBTSphy; variable que clasifica el tipo de objeto en la simulación
- cModule *Basestation; variable que guarda un puntero al modulo Basestation
- cModule *Mobile; variable que guarda un puntero al modulo Mobile

- int ms_moduleId; variable entera que guarda el id del objeto Mobile.Phy dentro del entorno de simulación OMNET++

Variables Trama de Subida (uplink)

- int rslotup; variable entera que indica el número de slots r
- double rslotup_dur; variable double que indica la duración en tiempo de un slot r
- int islotup; variable entera que indica el número de slots i de subida (i-slot-up)
- double islotup_dur; variable double que indica la duración en tiempo de un slot i de subida (i-slot-up)
- int frameup; variable entera que define el número de de tramas de subida generadas
- double frameup_dur; variable double que indica la duración en tiempo de una trama de subida

Variables Trama de Bajada (downlink)

- int aslotdw; variable entera que indica el número de slots a
- double aslotdw_dur; variable double que indica la duración en tiempo de un slot a (A-slot)
- int islotdw; variable entera que indica el número de slots i de bajada (i-slot-down)
- double islotdw_dur; variable double que indica la duración en tiempo de un slot i de bajada (i-slot-down)
- int framedw; variable entera que indica el número de tramas de bajada generadas
- double framedw_dur; variable double que indica la duración en tiempo de una trama de bajada

Variables Generales para las tramas de subida y bajada

- double k; variable double que almacena el desplazamiento R/A shift (variable de diseño)
- int m; variable entera que guarda el factor que relaciona el momento de inicio del slot A tomando como referencia el inicio del slot r
- int h; variable entera que guarda el desplazamiento i-slot offset (variable de diseño)
- int n; variable entera que indica el factor de duración del slot r en función del slot i de subida (variable de diseño)

Funciones

- cPhy(); función constructora de la clase cPhy
- ~cPhy(); función destructora de la clase cPhy
- initialize(); función inicializadora de la capa física
- handleMessage(); función manejadora de eventos
- finish(); función finalizadora de la capa física
- bts_init_phy(); función inicializadora de la capa física para la estación base
- ms_init_phy(); función inicializadora de la capa física para el terminal móvil
- bts_handleMessage_phy(); función manejadora de eventos de la capa física para la estación base
- ms_handleMessage_phy(); función manejadora de eventos de la capa física para el terminal móvil
- ms_read_file(); función que lee el archivo de texto mobile.txt, el cual contiene toda la información de los terminales móviles
- ms_tx_SIG_phy(); función usada por el terminal móvil para enviar el mensaje de SIG hacia la estación base
- ms_tx_PACKET_phy(); función usada por el terminal móvil para enviar los mensajes de PACKET / MESSAGE hacia la estación base
- bts_tx_ACK_phy(); función utilizada por la estación base para el envío de las notificaciones de aceptación tipo ACK hacia los terminales móviles cuyos mensajes de SIGs prosperaron

- `bts_tx_alert_phy()`; función usada por la estación base para el envío de las notificaciones hacia los terminales móviles que han logrado obtener un recurso de I-slot de subida
- `frame_up()`; función que configura la trama de subida
- `frame_down()`; función que configura la trama de bajada
- `debug_frame()`; función que habilita el debug en las tramas de subida y de bajada

Módulo Simple Capa MAC

Clase : `cMac`

Es una clase derivada de `cSimpleModule`

Define el comportamiento de la capa MAC para la estación base y el terminal móvil. Este módulo provee los algoritmos para la fase de canal y para la fase de planificación. Además que implementa el algoritmo de clase de servicio (QoS) que se necesita para clasificar el tráfico de datos en conversacional, interactivo, unidireccional y de fondo, y asignarle los i-slot-up que demananan

Variables para el Terminal Móvil

- `cQueue ms_local_queue`; variable tipo `cQueue` (OMNET++) que define la cola local local queue)
- `double bwt`; variable `double` del temporizador de ancho de banda (bandwidth timer)
- `bool ms_gen_status`; variable lógica para indicar el estado del generador de trafico
- `simtime_t ms_finActiveTime`; variable de tipo `simtime` (OMNET++) que predice la finalizacion del estadio ACTIVE
- `bool ms_tx_status`; variable lógica que define el estado del terminal móvil (disponible, ocupado)
- `bool ms_sig_status`; variable lógica que define el estado de la señal SIG (interna de la simulación)
- `bool ms_ack_status`; variable lógica que define el estado de la señal ACK (interna de la simulación)

- bool rslot_status; variable lógica que define el estado deL slot R (interna de la simulación)
- int tos; variable entera que indica el tipo de servicio; variable que clasifica el tipo de terminal como VOZ o DATOS
- int dropacktes; variable entera que guarda el conteo de los paquetes eliminados por el terminal móvil y debido al retardo en respuesta de la red
- struct granted, estructura que guarda un registro de los slots I de subida (i-slot-up) que fueron concedidos por la estación base al terminal móvil
- long msgLength; variable tipo long que almacena la longitud de Messages
- bool piggybacking; variable lógica que revisa el estado del piggybacking

Variables para la Estación Base

- bool aslot_status; variable de tipo lógica del estado del A-slot (interna de la simulación)
- cQueue bts_SIG_queue; variable de tipo cQueue, esta cola se usa almacenar los SIG enviados desde los terminales móviles a la estación base en cada intervalo de slot R. Esta cola se desocupa cuando se envían los ACKs a los Terminales Móviles
- cQueue bts_FCFS_queue; variable de tipo cQueue, esta cola se usa para almacenar los SIG enviados desde los Terminales Móviles a la estación base luego de haberse enviado los ACKs a los Móviles.
- cQueue bts_conversacional_queue, cQueue bts_interactivo_queue, cQueue bts_unidireccional_queue cQueue bts_RR_queue; variables de tipo cQueue, se usa para almacenar los REQs que envían los móviles de datos en distintas colas con diferente prioridad (QoS).
- struct bts_InfoMS, variable tipo estructura, usada para el almacenamiento temporal de mensajes REQs
- struct uplink, variable tipo estructura, usada para almacenar el estado de canales i-slot-up

- struct bts_scheduler, variable tipo estructura, constituye la base de datos que almacena información de los terminales móviles que han recibido el permiso para transmitir en uno o más i-slot-up.
- struct bts_statistic, variable tipo estructura, usada como registro de cuentas para propósitos estadísticos.
- int numMobiles; variable de tipo entera que toma el valor de la cantidad de terminales móviles creados durante la simulación
- int m; variable tipo entera, guarda el factor que relaciona el momento de inicio del slot a tomando como referencia el inicio del slot r
- int h; variable entera, guarda el valor i-slot offset
- int vislotup; variable entera que almacena el id del slot UP que asigna la estación base cuando se solicita un canal de voz desde el terminal móvil a través de un SIG
- int islotup; variable entera que almacena el I-slot UP asignado al terminal móvil
- int islotdown; variable entera que almacena el I-slot DOWN que alertara al terminal móvil
- int numIslotUP; variable entera que almacena la cantidad de i slot up que estan siendo utilizados

Variables Comunes

- const char *mNameMac; variable de tipo caracter que clasifica el tipo de modulo (BS o MS)
- bool isBTSmac; variable de tipo lógica que clasifica el tipo de objeto en la simulacion

Variables Estadísticas del terminal móvil

- int numSigsent; variable tipo entera que guarda el número de paquetes sig enviados
- int numAckrcvd; variable tipo entera que guarda el número de ack recibidos

- int numPktsent; variable tipo entera que guarda el número de paquetes/mensajes enviados por la capa MAC del Mobile
- cOutVector forward_Delay; variable tipo cOutVector que guarda el Vector para registrar los eventos forwardDelay
- bool fdelay; variable de tipo lógica, variable que decide si la cola local esta vacia - ms_local_queue
- int countFDelay; variable entera que guarda el número de veces que se cuenta el forwardDelay
- double acumforwardDelay; variable de tipo doble que acumula el total de tiempo originado por forwardDelay
- simtime_t forwardDelay; variable de tipo simtime que guarda el tiempo transcurrido desde el momento en que un mensaje ingresa a la cola local del terminal móvil (y que esta vacía) hasta que inicia el siguiente slot R
- cOutVector queuing_Delay; variable de tipo cOutVector que registra los eventos queuingDelay
- int countQDelay; variable de tipo entera que guarda el número de veces que se cuenta el queuingDelay
- double acumqueuingDelay; variable de tipo doble que acumula el total de tiempo originado por queuingDelay
- simtime_t queuingDelay; variable de tipo simtime que guarda el tiempo transcurrido desde el momento en que un mensaje ingresa a la cola local del Mobile (.y que NO esta vacía) hasta que alcanza la primera posición de la cola local
- cOutVector alloc_Delay; variable de tipo cOutVector que registra los eventos allocDelay
- bool adelay; variable de tipo lógica que decide el inicio de captura estadística de ALLOC DELAY
- int countADelay; variable de tipo entera que guarda el número de veces que se cuenta el allocDelay
- double acumallocDelay; variable de tipo doble que acumula el total de tiempo originado por allocDelay

- simtime_t allocDelay; variable de tipo simtime que guarda el Tiempo transcurrido desde el momento en que finaliza el periodo de forward hasta el momento en que la BS autoriza al MS a transmitir una celda (message / packet)
- cOutVector RR_Delay; variable de tipo cOutVector que registra los eventos RRDelay
- int countRRDelay; variable de tipo entera que guarda el número de veces que se cuenta el RRDelay
- double acumRRDelay; variable de tipo double que acumula el total de tiempo originado por RRDelay
- simtime_t RRDelay; variable de tipo simtime que guarda el tiempo necesario para la transmisión completa de un mensaje desde el momento en que es insertado en la cola RR hasta el momento en que es transmitido
- int aggregateThroughput; variable de tipo entera que calcula el Promedio de bits transmitidos exitosamente por todos los terminales móviles por unidad de tiempo
- int avgQueueOccupancy; variable de tipo entera que guarda el Promedio de packets (messages) encolados en la cola local
- int utilizationFactor; variable de tipo entera que guarda la Fracción de I-slots usados para transmitir información del usuario

Variables Estadísticas de la estación base

- simtime_t txDelay; variable de tipo simtime que guarda el Tiempo transcurrido desde el momento en que el packet (message) es insertado en la cola FCFS (RR) de la estación base hasta el momento en que es atendido (eliminado)
- cOutVector access_DelayVoice; variable de tipo cOutVector que registra los eventos accessDelay para Voice
- int countAccessDelayVoice; variable de tipo entera que guarda el número de veces que se cuenta el accessDelay para Voice

- double acumAccessDelayVoice; variable de tipo doble que acumula el total de tiempo originado por accessDelay para Voice
- simtime_t accessDelayVoice; variable de tipo simtime que guarda el tiempo necesario para la transmisión completa del packet desde el momento en que ha sido generado por la capa de tráfico
- cOutVector access_DelayMessage; access_DelayConversacional, access_DelayInteractivo, access_DelayUnidireccional, variable de tipo cOutVector que registra los eventos accessDelay para Data
- int countAccessDelayMessage; variable entera que guarda el número de veces que se cuenta el accessDelay para Data
- double acumAccessDelayMessage; variable de tipo doble que acumula el total de tiempo originado por accessDelay para Data
- simtime_t accessDelayMessage; variable de tipo simtime que guarda el Tiempo necesario para la transmisión completa del mensaje y desde el momento en que ha sido generado por la capa de tráfico
- cLongHistogram slot_Utilization; variable de tipo cLongHistogram, es un Histograma que almacena la ocupación de los I-slots
- int slotutilization; variable entera que almacena temporalmente el total de pkts recibidos por trama

Funciones

- cMac();función constructora de la capa MAC
- ~cMac();función destructora de la capa MAC
- initialize();función de inicialización de la capa MAC
- handleMessage();función manejadora de eventos de la capa MAC
- finish(); función finalizadora de la capa MAC
- bts_handleMessage_mac(); función manejadora de eventos para la estación base
- ms_handleMessage_mac();función manejadora de eventos del terminal móvil ms_tx_SIG_mac();función que transmite el paquete SIG a la capa física

- `ms_writeDB_mac()`;función que guarda en una DB información de asignacion de I slots UP
- `ms_clearDB_mac()`;función que limpia la base de datos que contiene información de los I slots UP asignados
- `ms_chanbwrsvph_v_mac()`;función con el algoritmo de reserva de canal para voz(channel bandwidth reservation phase)
- `ms_chanbwrsvph_d_mac()`;función con el algoritmo de reserva de canal para datos(channel bandwidth reservation phase)
- `bts_analyze_SIG_mac()`;función que analiza los SIG recibidos, clasifica el SIG en función del tipo de tráfico y asigna el i-slot-up tomando en cuenta la prioridad: conversacional, interactivo, unidireccional y de fondo. Esta función provee la calidad de servicio al protocolo mac.
- `bts_tx_ACK_mac()`; función que transmite la señal de ACK hacia el terminal móvil
- `bts_decode_SIG_mac()`;función para decodificar el SIG
- `bts_bwmanagement_v_mac()`;función que administra el ancho de banda para voz
- `bts_scheduler_mac()`;función que programa la notificación de disponibilidad del recurso I-slot UP
- `bts_writeDB_mac()`;función que guarda en una DB información de asignacion de I slots UP
- `bts_updateDB_mac()`;función que actualiza la DB I slots UP asignados luego que el terminal móvil le envia el ultimo paquete de la cola local
- `bts_clearDB_mac()`; función que limpia la DB I slots UP asignados
- `bts_alertallocation_mac()`;función que alerta al terminal móvil acerca del I-slot UP asignado por la estación base
- `bts_updateSTAT_mac()`;función que actualiza las ESTADISTICAS luego que recibe un paquete exitosamente del terminal móvil
- `bool bts_thereIsResource_mac()`;función que busca unicamente si existe o no I-slot up disponibles. Retorna un valor booleano

- `bts_scanRR_mac()`; función que scanea las colas RRs y asigna los recursos de I-slots
- `bts_updateQueue_mac()`; función para actualizar/eliminar la cola RR creada para el terminal móvil

Módulo Simple Capa de Tráfico

Clase : `cTraffic`

Es una clase derivada de `cSimpleModule`

Define el comportamiento de la capa de tráfico para la estación base y el terminal móvil. En el terminal móvil se comporta como un generador de tráfico, en la estación base se comporta como un colector ó sumidero. Permite generar los siguientes tipos de tráfico: voz, juego en línea (conversacional), búsqueda web www (interactivo), video streaming (unidireccional) y tráfico de datos como el usado en [2].

Variables Generales

- `cMessage *startStopBurst`; variable de tipo `cMessage`, usada para indicar el INICIO/PARADA del generador de tráfico (BURST)
- `cMessage *sendMessage`; variable de tipo `cMessage`, usada para iniciar el ENVIO de trafico (BURST)
- `bool isbusy`; variable de tipo lógica, usada para indicar el estado del generador de trafico.
- `bool firstSent`; variable de tipo lógica que permite hacer una transmisión cada vez que se ingresa al estado Active y proveniente del estado Sleep
- `int msgsAcum`; variable de tipo entera que cuenta el total de mensajes enviados
- `int msgsPar`; variable de tipo entera que cuenta los mensajes enviados durante el estado ON en particular
- `int tos`; variable de tipo entera que indica el tipo de servicio (tos), especifica si el trafico a generarse será de voz o data

- int onTimes; variable de tipo entera que cuenta las veces que el generador ingreso al estado ON
- int offTimes; variable entera que cuenta las veces que el generador ingresó al estado OFF
- simtime_t totalActiveTime; variable de tipo simtime que almacena el tiempo que el generador permaneció ACTIVE
- simtime_t creationTime; variable de tipo simtime que almacena el tiempo de creación del paquete/mensaje
- simtime_t finActiveTime; variable de tipo simtime que almacena el valor de tiempo en que culminara el estadio ACTIVE
- cPar *msgLength; variable de tipo cPar usada para especificar la longitud en bits del mensaje/paquete (voz o data) enviado
- const char *mNameTrf; variable de tipo caracter que clasifica el tipo de modulo (BS o MS)
- bool isBTStrf; variable de tipo lógica que clasifica el tipo de objeto en la simulación

VARIABLES PARA EL GENERADOR DE VOZ

- double meanoff; variable de tipo doble usado en distribución exponencial para indicar la media "mean" del estadio OFF
- double meanon; variable de tipo doble usado en distribución exponencial para indicar la media "mean" del estadio ON
- double interarrival; variable de tipo doble, usada para indicar la velocidad de generación de paquetes dentro del estadio ON

VARIABLES PARA EL GENERADOR DE DATOS

- double meaninterarrival; variable de tipo doble, toma el valor "mean" de la distribución exponencial para configurar el tiempo entre arribo dentro del estadio ON
- double offscale; variable de tipo doble, en la distribución Weibull, indica el parámetro "scale" del estadio OFF

- double offshape; variable de tipo doble, en la distribución Weibull, indica el parámetro "shape" del estadio OFF
- double onscale; variable de tipo doble, en la distribución Weibull, indica el parámetro "scale" del estadio ON
- double onshape; variable de tipo doble, en la distribución Weibull, indica el parámetro "shape" del estadio ON

Variables Estadísticas

- cOutVector numpktsActiveVector; variable de tipo cOutVector que almacena el número de paquetes generados en estado activo
- cLongHistogram numpktsCountStats; variable de tipo cLonhHistogram que almacena el valor escalar de la cantidad de paquetes generados
- cLongHistogram activetimeCountStats; variable de tipo cLonhHistogram que almacena el valor escalar de la cantidad de veces que el generador pasó al estado de activo

El algoritmo de estados finitos (FSM – Finite State Machine)

El algoritmo de estados finitos es el corazón del generador de tráfico. Los estados permitidos son: Inicialización, Apagado, Activo y Transmitiendo.

Usa una variable de la clase cFSM definida en la librería de OMNET++.

Funciones:

- cTraffic(); función constructora de la clase derivada cTraffic
- ~cTraffic(); función destructora de la clase derivada cTraffic
- initialize(); función que inicializa la clase cTraffic
- handleMessage(); función manejadora de eventos de la clase cTraffic
- finish(); función que finaliza de la clase derivada cTraffic
- bts_handleMessage_trf(); función manejadora de eventos de la estación base, se comporta como un sumidero de eventos
- ms_handleMessage_trf(); función manejadora de eventos del terminal móvil, se comporta como un generador de tráfico
- sending(); función que envía mensajes de tipo protocol a la capa MAC

2.3.3 Parámetros de la Simulación

Como se explicó al inicio, el protocolo MAC eficiente para redes inalámbricas brinda una mejora al protocolo CRDA, que consiste en la modificación del algoritmo de planificación (scheduling), aplicando políticas de calidad de servicio (QoS). Para validar el comportamiento del protocolo CRDA que se programó usando OMNET++ se tuvieron que usar inicialmente los parámetros sugeridos en [2]. Una vez obtenidos los resultados similares a CRDA, se pasó a la siguiente fase que consistió en el reemplazo de las fuentes de tráfico de datos en cuatro categorías diferentes y la inserción del nuevo algoritmo planificación con soporte de QoS.

Parámetros Operacionales para todas las simulaciones

Tabla 2.3 Parámetros Operacionales del Protocolo MAC Eficiente

Duración de la Trama (subida ó bajada): Frame-up-duration Frame-down-duration	4.9968 ms
Duración de un slot de información: I-slot-up-duration I-slot-down-duration	69.4us
Número de slots que conforman una trama:	72 slots
Número de slots de Información por trama: #I-slot-up #I-slot-down)	68 slots
Longitud del R-slot expresado en unidades de I-slots n (fórmula 2.5)	4 slots
Longitud del A-slot expresado en unidades de I-slots	4 slots
R/A shift m (fórmula 2.12)	4 slots
I-slot-offset (h)	2 slots
Carga útil del I-slot –up/down	62 bits
Capacidad de transmisión para el canal	843.2 kbit/s
Probabilidad de colisión en R.-slot	0.01

Nota 2.1:

Algunos de los valores de la Tabla 2.3 fueron sacados de la referencia [2]

Parámetros para Voz en todas las simulaciones

Tabla 2.4 Parámetros para Voz

Modelo de Tráfico para Voz	ON/OFF
Distribución Estadística del Período ON	Exponencial
Promedio del Período ON	1.00 segundo
Distribución Estadística del Período OFF	Exponencial
Promedio del Período OFF	1.35 segundos
Distribución del Período entre arribo en estado ON	Constante cada 4.9968 ms
bwt	10 ms

Nota 2.2:

Los valores de la Tabla 2.4 fueron sacados de la referencia [2]

Parámetros para Tráfico de Fondo: datos sin prioridad

Tabla 2.5 Parámetros para Datos sin prioridad

Modelo de Tráfico para Datos	On/OFF
Distribución Estadística del Período ON	Weibull
Promedio de la duración en ON: - Parámetro de Forma - Parámetro de Escala	0.88 3.10
Distribución del tiempo entre arribo durante el estado ON	Exponencial
Valor medio de la distribución del tiempo entre arribo (λ)	Ver nota 2.3
Promedio de la duración en OFF: - Parámetro de Forma - Parámetro de Escala	0.88 21.40
bwt	deshabilitado
Longitud del mensaje de datos: - Constante	6 celdas

Nota 2.3:

- Algunos de los valores de la Tabla 2.5 fueron sacados de la referencia [2]
- Debido a que la distribución del tiempo entre arribo durante el estado de encendido (ON), se modeló usando una distribución exponencial, se hace necesario definir el valor medio de dicha distribución. Al valor medio de una distribución exponencial se le conoce como λ , dicho valor no ha sido

especificado explícitamente por [2]. De esta manera se buscó un valor de λ , que proporcione los resultados más cercanos a [2]. Ver capítulo 3, Primer Grupo de Pruebas, subgrupo #2.

Parámetros para Tráfico Conversacional: juegos en línea

Tabla 2.6 Parámetros para clase conversacional

Parámetro	Valor	Comentario
Tiempo medio de arribo de llamada de paquetes	5 segundos	Distribución exponencial
Tiempo medio de lectura	5 segundos	Distribución exponencial
Tamaño del datagrama	576 bytes ó 75 celdas	Fijo, constante
Media de tiempo entre arribo de los datagramas	40 ms	Distribución Log-normal, desviación estándar de 40 ms

Nota 2.4:

Los valores de la Tabla 2.6 fueron sacados de la referencia [20]

Parámetros para Tráfico Interactivo: búsqueda web

Tabla 2.7 Parámetros para clase interactiva

Parámetro	Valor	Comentario
Tiempo medio de arribo de cada conexión	Media : 0.5967 Desviación Estándar 1.62215906	Distribución LogNormal
Volumen de datos por conexión (bytes)	Media : 7.4343 Desviación Estándar 1.86316934	Distribución LogNormal
Media de tiempo entre arribo de los paquetes	Media : -3.2441 Desviación Estándar 2.12454701	Distribución LogNormal

Nota 2.5:

Los valores de la Tabla 2.7 fueron sacados de la referencia [22]

Parámetros para Tráfico Unidireccional: video streaming

Tabla 2.8 Parámetros para clase unidireccional

Tipos de Información	Tiempo entre arribo entre el inicio de cada frame	Número de paquetes en un frame	Tamaño del paquete	Tiempo entre arribo de paquetes dentro de un frame
Distribución	Determinístico (Basado en 10fps)	Determinístico	Pareto acotado (Media= 100bytes, Max= 250bytes)	Pareto acotado (Media= 6ms, Max= 12.5ms)
Parámetros de Distribución	100ms	8	K = 40bytes $\alpha = 1.2$	K = 2.5ms $\alpha = 1.2$

Nota 2.6:

Los valores de la Tabla 2.8 fueron sacados de la referencia [20]

2.3.4 Restricciones del Trabajo de Tesis

El trabajo de tesis se desarrolló y probó bajo las siguientes restricciones que deben tomarse en consideración:

- El entorno simula una red inalámbrica celular ad-hoc
- No se tomaron en cuenta las colisiones al considerarse que se utiliza codificación CDMA en el canal de subida
- No se simuló el canal radioeléctrico con las contribuciones de ruido, errores aleatorios, desvanecimiento de la señal por la trayectoria. Básicamente se asumió que todos los terminales móviles tenían acceso a la estación base y con la misma intensidad de señal.
- No se simuló el tráfico en el canal de bajada, desde la estación base a los terminales móviles. Únicamente se cursó tráfico por señalización (alertas por aceptación de tráfico de subida) en ese tramo.
- El trabajo tiene por objeto probar la asignación de recursos (canales de subida) realizada por el protocolo MAC eficiente, cuando recibe tráfico que demanda diferentes clases de QoS, tales como el tráfico de voz, conversacional, interactivo, unidireccional y de fondo.

CAPÍTULO 3

PRUEBAS REALIZADAS Y RESULTADOS OBTENIDOS

Como se comentara al inicio, el protocolo MAC eficiente de la presente tesis basa su funcionamiento en CRDA, [2] agregándole mejoras al algoritmo de planificación (scheduling), a fin de que pueda ser capaz de transportar tráfico con distintas calidades de servicio.

En ese sentido las pruebas realizadas consistieron en:

- Primer grupo de pruebas, sirvieron para validar el funcionamiento del protocolo CRDA que se programó en OMNET++. El objetivo fue comparar nuestros resultados con los de la referencia [2] y comprobar el diseño inferido (sección 2.2) a través de los modelos de diagramas de flujo, diagramas de señalización y declaraciones de fórmulas. En este grupo de pruebas se tomaron los parámetros de simulación que se usaron en [2] (Tablas 2.3, 2.4 y 2.5).
- Segundo grupo de pruebas, sirvieron para ejecutar el protocolo CRDA con los nuevos generadores de tráfico (conversacional, interactivo, unidireccional y de fondo) que demandan distintos requerimientos en QoS. El objetivo fue analizar el desempeño de CRDA en escenarios que demandan distintos requerimientos de QoS. En este grupo de pruebas se tomaron los parámetros de simulación de las Tablas 2.3 – 2.8.
- Tercer grupo de pruebas para ejecutar nuestro protocolo MAC eficiente (llamado ECRDA), con los nuevos generadores de tráfico con el objeto de analizar el desempeño y compararlo con CRDA. En este grupo de pruebas se tomaron los parámetros de simulación de las Tablas 2.3 – 2.8.

El listado de las simulaciones realizadas se presenta en el Anexo E – Software de la Tesis, sección E.2; y también se adjunta un CD con los programas fuente y las instrucciones para ejecutar las simulaciones de este capítulo.

3.1 Requerimientos computacionales en la simulación

Las simulaciones se realizaron usando 6 computadoras del Laboratorio de Postgrado FIEE (UNI) y cuyas características de hardware y software son las que se muestran a continuación:

Requerimientos de Hardware de las computadoras donde se ejecutaron las simulaciones:

- Procesador Intel Core 2 Duo E8200 de 2.66Ghz
- Memoria RAM de 4GBytes

Requerimientos de Software de las computadoras donde se ejecutaron las simulaciones:

- Sistema Operativo Open Suse 11., kernel 2.6.27.7-9-pae
- Framework del Simulador OMNET++ versión 3.3p1
- Compilador GNU C++ del Sistema gcc-c++ versión 4.3-34.168(i586)
- Compilador GNU de C++ gcc43-c++ version 4.3.3_20081022-9.5(i586)
- Librería GNU de C glibc versión 2.9-2.3(i586)

3.2 Pruebas realizadas y Resultados obtenidos

En todas las pruebas el escenario consiste de una estación base y una cierta cantidad de terminales móviles que se encuentran enviando tráfico hacia la estación base. El tráfico enviado proviene de aplicaciones como voz en paquetes, juegos en línea, búsqueda web, video streaming y datos sin prioridad. En ese sentido nuestro escenario prueba el desempeño del protocolo en el segmento del canal de subida (uplink).

3.2.1 Primer grupo de pruebas

Estas pruebas se realizaron a fin de validar que el protocolo CRDA programado en OMNET++ responda con resultados similares a [2] y valide los modelos inferidos en nuestro diseño. La principal desventaja es que los autores de [2] no han especificado algunos detalles importantes como por ejemplo:

- el parámetro “valor medio de la distribución del tiempo entre arribo” (parámetro λ de la Tabla 2.5). En [2] solo indican que han usado una distribución exponencial pero no detallan el valor de λ , que es necesario para simular la distribución.
- En [2] no se indica el tiempo de la simulación ni la cantidad de simulaciones que realizaron para conseguir sus resultados. En ese sentido, a fin de lograr resultados confiables se creyó conveniente simular las distribuciones de tráfico y comparar los resultados con los valores promedio esperados (Promedio Periodo ON, Promedio Periodo OFF – Tablas 2.4 y 2.5).

De esta manera el grupo de pruebas realizadas consistió en:

Primer Grupo, Subgrupo #1 - Pruebas de las fuentes de tráfico y tiempo óptimo de simulación

Estas pruebas tuvieron por objetivo capturar la naturaleza estadística de las fuentes de tráfico y determinar el tiempo óptimo de simulación. En ese sentido se realizaron varias ejecuciones con el objeto de verificar que los parámetros usados estén dentro de los valores promedios proporcionados en la sección 2.3.3 - Parámetros de la Simulación. Se probaron los siguientes escenarios:

- **Prueba 1001:** Simulación de 01 móvil de voz y 01 móvil de datos durante 27,000 segundos
- **Prueba 1002 :** Simulación de 80 móviles de voz y 40 móviles de datos durante 1,000 segundos
- **Prueba 1003 :** Simulación de 80 móviles de voz y 40 móviles de datos durante 500 segundos
- **Prueba 1004 :** Simulación de 80 móviles de voz y 40 móviles de datos durante 200 segundos
- **Prueba 1005 :** Simulación de 80 móviles de voz y 40 móviles de datos durante 100 segundos

Primer Grupo, Subgrupo #1 – Resultados

Tabla 3.1 Pruebas con las fuentes de tráfico y tiempo de simulación

Caso	TS	Terminales de Voz – Promedio expresado en segundos				Terminales de Datos – Promedio expresado en segundos			
		Estado Encendido		Estado Apagado		Estado Encendido		Estado Apagado	
		VO	VE	VO	VE	VO	VE	VO	VE
01V, 01D - Prueba 1001	27,000	1.0624	1.00	1.3593	1.35	2.8758	3.30	21.6468	22.80
80V, 80D – Prueba 1002	1,000	1.0625	1.00	1.3608	1.35	2.9292	3.30	22.9705	22.80
80V, 80D - Prueba 1003	500	1.0660	1.00	1.3625	1.35	2.9230	3.30	21.4367	22.80
80V, 80D – Prueba 1004	200	1.0077	1.00	1.3710	1.35	2.7789	3.30	21.1801	22.80
80V, 80D - Prueba 1005	100	1.0689	1.00	1.3748	1.35	2.5593	3.30	20.9637	22.80

Leyenda:

TS: Tiempo de simulación expresado en segundos

VO: Valor obtenido

VE: Valor esperado

De la Tabla 3.1 puede observarse:

- La métrica usada fue el promedio de duración de los periodos de encendido (ON) y apagado (OFF) de las fuentes de tráfico para voz y datos,
- Para el caso de voz, la diferencia entre el valor obtenido y el valor esperado, es del orden de las centésimas, sin embargo, el valor obtenido tiende a estabilizarse a partir de los 500 segundos de simulación.

- Para el caso de datos, la diferencia entre el valor obtenido y el valor esperado, es del orden de las décimas y con tendencia a reducirse más a partir de los 500 segundos de simulación. .

Se puede inferir:

- Los valores obtenidos son similares a los valores esperados, obteniéndose mayor precisión con las fuentes de voz. Con las fuentes de datos se mejora la precisión cuando se aumenta el tiempo de simulación.
- A partir de 500 segundos de simulación, la naturaleza estocástica de las fuentes de tráfico tiende a ser similar y ha estabilizarse.

Primer Grupo, Subgrupo #2 - Pruebas para hallar un valor óptimo de λ

El parámetro λ se define como el valor medio de la distribución exponencial que para este caso, modela el tiempo entre arribo de paquetes, durante el estado de encendido (ON). El segundo grupo de simulaciones se realizó con el objeto de ajustar el parámetro λ a un valor óptimo. En este punto se fijó en realizar las pruebas con 100 terminales de voz y 40 terminales de datos. Además, de los resultados obtenidos en Primer Grupo, Subgrupo #1, se estableció un tiempo de simulación de 1000 segundos para estas pruebas y las posteriores. Los escenarios probados fueron:

- **Prueba 1006** : Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de $\lambda=0.0100$ exponencial
- **Prueba 1007** : Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de $\lambda=0.0095$ exponencial
- **Prueba 1008** : Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de $\lambda=0.0090$ exponencial
- **Prueba 1009** : Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de $\lambda=0.0087$ exponencial
- **Prueba 1010** : Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de $\lambda=0.0085$ exponencial
- **Prueba 1011** : Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de $\lambda=0.0080$ exponencial

Primer Grupo, Subgrupo #2 – Resultados

En este segundo grupo de simulaciones se tomaron medidas del Promedio por Retardo de Acceso para Datos, que mide el tiempo necesario para la transmisión completa del mensaje, desde el instante en que el mensaje se inserta en la cola local del terminal móvil, hasta que es entregado a la estación base. El valor capturado por los colectores estadísticos recoge el promedio de todas las muestras que se generaron desde el inicio hasta el final de la simulación.

Tabla 3.2 Resultado de las pruebas para hallar un valor óptimo de λ

Caso	Tiempo de simulación (seg)	Terminales de Datos – Promedio por Retardo de Acceso para Datos expresado en segundos		Terminales de Voz – Promedio por Retardo de Acceso para Voz expresado en segundos
		Valor medio de la distribución exponencial (λ)	Valor Obtenido	Valor Obtenido
100V, 40D – Prueba 1006	1,000	0.0100	0.24651491	0.00447317
100V, 40D – Prueba 1007	1,000	0.0095	0.26214952	0.00447085
100V, 40D – Prueba 1008	1,000	0.0090	0.42438525	0.00446049
100V, 40D – Prueba 1009	1,000	0.0087	0.58058399	0.00445728
100V, 40D – Prueba 1010	1,000	0.0085	0.82781357	0.00446542
100V, 40D – Prueba 1011	1,000	0.0080	1.49294394	0.00447614
100V, 40D	No se especifica	No se especifica	0.62 aprox, [2].	No se especifica

De lo obtenido en la Tabla 3.2, referente al Promedio por Retardo de Acceso en los terminales de Voz, puede observarse que dicho valor se mantiene casi invariable en las seis simulaciones, esto valida lo comentado en [2]: “el protocolo CRDA desacopla eficientemente los flujos de voz y datos, priorizando las fuentes de voz sobre las fuentes de datos”.

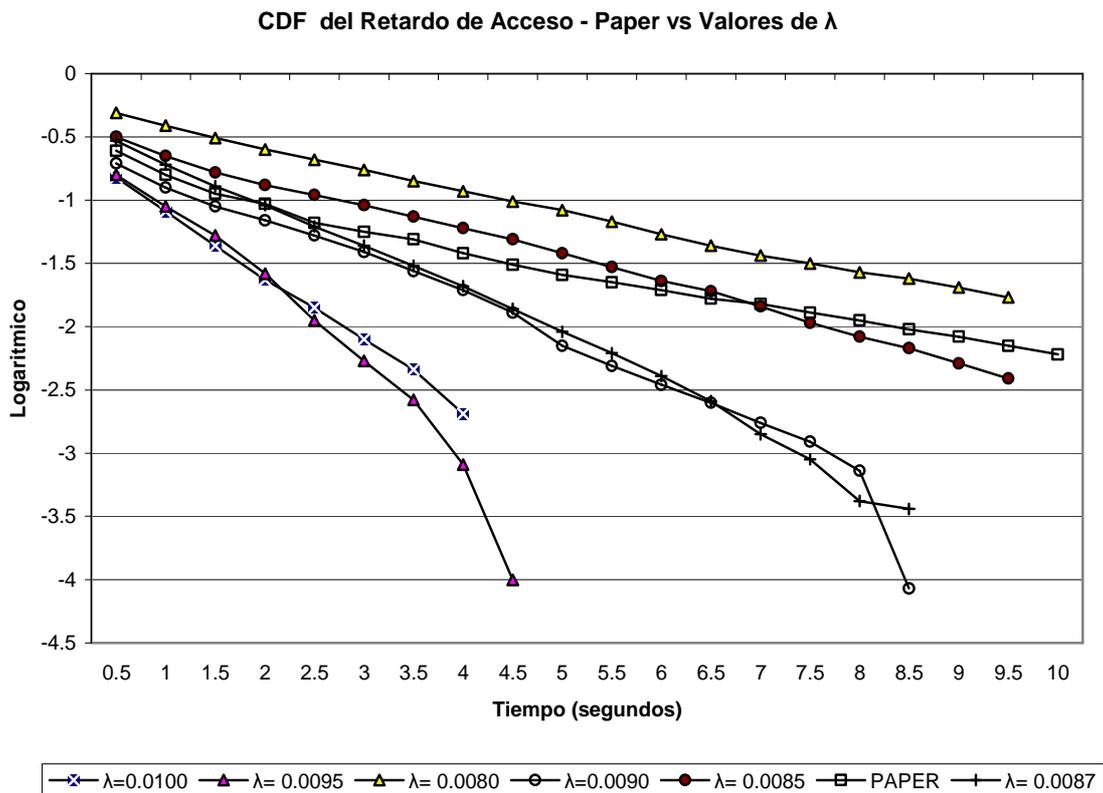


Figura 3.1 CDF del Retardo de Acceso, compara el CDF del paper, [2] vs diferentes valores de λ , para una simulación de 140 terminales: 100V y 40D

Con referencia a los terminales de datos, de la Tabla 3.2 se observa que a valores pequeños de λ la naturaleza estocástica de las fuentes de tráfico tiende a ser más persistente (heavy tailed) y el promedio por Retardo de Acceso tiende a incrementarse enormemente. Por otro lado, a valores altos de λ se obtiene un retardo de acceso muy inferior a los reportados en [2] (ver la última fila de la Tabla 3.2). De acuerdo a lo reportado en [2] el valor debería rondar por 0.62, de esta

manera puede observarse que el valor de λ puede estar entre 0.0087 y 0.0085. Para una mejor elección del valor de λ es conveniente evaluar la CDF por Retardo de Acceso para distintos valores de λ , y compararlo con la curva CDF reportada en [2]. La figura 3.1 muestra los resultados de la curva CDF para distintos valores de λ y que varían discretamente entre 0.0080 y 0.0100. Se observa que para un valor λ de 0.0085 la curva CDF es semejante a lo reportado en [2], tomando en cuenta el escenario de simulación con 140 terminales móviles: 40 de datos y 100 de voz. De esta manera se escogió el valor referencial de λ igual a 0.0085.

Primer Grupo, Subgrupo #3 – Pruebas de validación del Promedio por Retardo de Acceso

La métrica usada fue el Promedio por Retardo de Acceso (PRA). Para cada escenario se corrieron hasta 8 simulaciones y cada simulación se configuró en 1000 segundos y un valor de λ igual a 0.0085. Los resultados de esta prueba sirvieron para validar que nuestro protocolo CRDA programado en OMNET++ responde con la misma tendencia estadística que en [2].

- **Prueba 1012** : Simulación de 100 móviles de voz y 20 móviles de datos
- **Prueba 1013** : Simulación de 100 móviles de voz y 30 móviles de datos
- **Prueba 1014** : Simulación de 100 móviles de voz y 35 móviles de datos
- **Prueba 1015** : Simulación de 100 móviles de voz y 40 móviles de datos
- **Prueba 1016** : Simulación de 100 móviles de voz y 42 móviles de datos
- **Prueba 1017** : Simulación de 100 móviles de voz y 44 móviles de datos

Primer Grupo, Subgrupo #3 – Resultados

Tabla 3.3 Promedio por Retardo de Acceso para Terminales de Datos

Caso	Promedio por Retardo de Acceso para Terminales de Datos (Paper - [2])	Promedio por Retardo de Acceso para Terminales de Datos (Valor Obtenido)
100V, 20D – Prueba 1012	0.001000	0.032522453
100V, 30D – Prueba 1013	0.110000	0.307799935
100V, 35D – Prueba 1014	0.250000	0.529823221
100V, 40D – Prueba 1015	0.650000	0.951177327
100V, 42D – Prueba 1016	0.950000	1.377865864
100V, 44D – Prueba 1017	1.450000	2.721558114

Para un mejor análisis, la Tabla 3.3 se esquematiza en la figura 3.2, y se observa que el protocolo CRDA programado en OMNET++ sigue la tendencia estadística de la referencia [2]. También se observa que el aumento de terminales de datos más allá de 20 unidades produce un aumento exponencial del Promedio por Retardo de Acceso (PRA). Las diferencias entre los valores obtenidos versus los valores de [2], se deben a que, se han ejecutado las simulaciones con un valor de λ cercano al utilizado en [2]. Con los resultados obtenidos en la Tabla 3.3 se comprueba que, el protocolo CRDA programado en OMNET++, tiene un comportamiento estadístico semejante al reportado en [2]; con lo cual se validan los modelos de diseño inferidos en esta tesis y se comprueba que OMNET++ es una herramienta confiable para la simulación de eventos discretos.

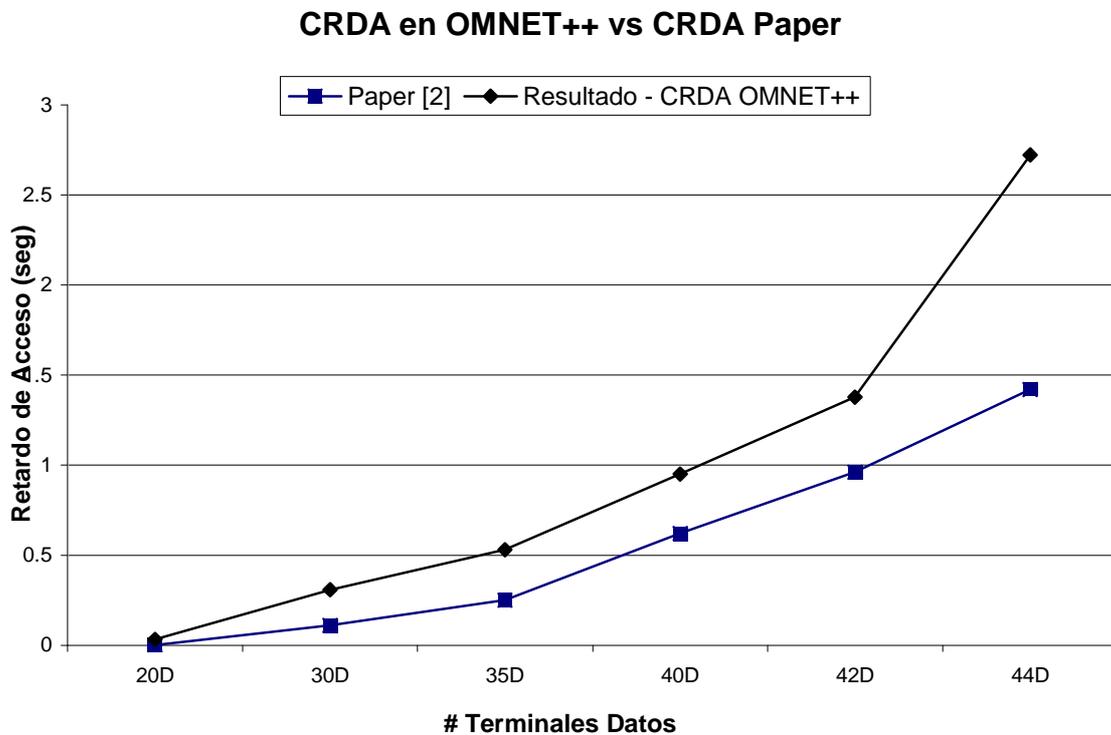


Figura 3.2 Promedio por Retardo de Acceso para Terminales de Datos - Simulación vs [2]

3.2.2 Segundo grupo de pruebas

El objetivo de estas pruebas fue averiguar el desempeño de CRDA cuando se usan fuentes de tráfico con distintos requerimientos de calidad de servicio (QoS). Para estas pruebas la métrica que se usó fue el valor del Promedio por Retardo de Acceso (PRA). Las simulaciones se ejecutaron durante 500 segundos (tiempo de simulación sugerido en [20]) y los escenarios probados fueron:

- **Prueba 1018** : Simulación de 100 móviles de voz, 40 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 25 de clase de fondo.
- **Prueba 1019** : Simulación de 100 móviles de voz, 20 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 5 de clase de fondo.

Segundo Grupo – Resultados

Tabla 3.4 CRDA / Desempeño con fuentes de tráfico que demandan distintos requerimientos de QoS - Promedio por Retardo de Acceso (PRA)

MAC	PRA CONVERSACIONAL (seg)	PRA INTERACTIVO (seg)	PRA UNIDIRECCIONAL (seg)	PRA FONDO (seg)
Umbral [19]	<< 1	~ 1	< 10	> 10
CRDA Prueba 1018	0.39687436	0.77915371	59.94646301	6.75392313
CRDA Prueba 1019	0.16005208	0.32900324	57.62284024	1.78231558

En la Tabla 3.4 se comparan los resultados de las pruebas de CRDA con las nuevas fuentes de tráfico, tomando como referencia de umbral los valores sugeridos en [19]. El menor retardo se presenta en el escenario con 120 terminales, siendo el tráfico de la clase unidireccional el más afectado. Al inicio se mencionó que CRDA trata a todos los tráficos con justicia; aparentemente esta situación no se refleja en los resultados pues se podría inferir que el orden de prioridad de atención es el siguiente: tráfico conversacional, interactivo, de fondo, unidireccional. Esta situación se debe a que la aplicación de tráfico unidireccional (video streaming) es muy persistente, en comparación a las otras aplicaciones, y genera un tráfico muy alto, lo cual dispara los valores obtenidos a resultados muy altos en comparación al resto, y no deja observar con claridad que CRDA trata a todos los tráficos equitativamente. Sin embargo al reducir el número de terminales móviles a 120, específicamente al reducir solo los terminales con tráfico de fondo, vemos que la diferencia entre los valores PRA para los tipos de tráfico conversacional, interactivo y de fondo se va reduciendo más. Inclusive los valores de PRA del tráfico unidireccional no se ven afectados significativamente por la reducción de terminales, validando que CRDA trata a todos los tráficos equitativamente.

Tabla 3.5 CRDA / Desempeño con fuentes de tráfico que demandan distintos requerimientos de QoS – Valor Pico por Retardo de Acceso

MAC	PICO CONVERSACIONAL (seg)	PICO INTERACTIVO (seg)	PICO UNIDIRECCIONAL (seg)	PICO FONDO (seg)
Umbral [19]	<< 1	~ 1	< 10	> 10
CRDA Prueba 1018	1.91303962	7.02496390	147.65103734	38.58589180
CRDA Prueba 1019	0.54329386	2.46191365	186.35455077	12.24412053

Otro comentario que podemos hacer a la Tabla 3.4 es que, aparentemente CRDA cumple con los umbrales establecidos para los tráficos interactivo y de fondo. Para hacer un análisis más profundo se analizaron los valores pico de PRA, obtenidos en este grupo de simulaciones. Los resultados se presentan en la Tabla 3.5, donde se observa que para todos los casos los valores están muy por encima del umbral referencial. De esta manera se comprueba que CRDA no ofrece soporte de QoS. Si se desea lograr mejores resultados es conveniente probar escenarios con una cantidad de terminales móviles menor a la reportada en la Prueba 1019 (menor a 120), por lo tanto CRDA logrará un mejor desempeño al reducir su capacidad de atención.

3.2.3 Tercer grupo de pruebas

Para este grupo de pruebas se simuló el desempeño del protocolo MAC eficiente (llamado ECRDA), en presencia de fuentes de tráfico con distintos requerimientos de QoS. La métrica usada fue el valor del Promedio por Retardo de Acceso. Las simulaciones se ejecutaron durante 500 segundos de simulación y los escenarios probados fueron:

- **Prueba 1020** : Simulación de 100 móviles de voz, 40 móviles de datos: 10 de clase conversacional, 10 de clase interactiva, 10 de clase unidireccional y 10 de clase de fondo.

- **Prueba 1021** : Simulación de 100 móviles de voz, 40 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 25 de clase de fondo.
- **Prueba 1022** : Simulación de 100 móviles de voz, 20 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 5 de clase de fondo.

Tercer Grupo – Resultados

El protocolo MAC eficiente ECRDA, cuenta con una política de priorización para que asigne los canales i-slot-up en el siguiente orden: primero el tráfico de voz, segundo el tráfico conversacional, tercero el tráfico interactivo, cuarto el tráfico unidireccional y quinto el tráfico de fondo. El objetivo de estas pruebas fue comparar el desempeño de CRDA versus nuestro protocolo MAC eficiente (ECRDA), bajo los términos de calidad de servicio. La métrica de evaluación que usamos fue el PRA (Promedio por Retardo de Acceso) para las distintas fuentes de tráfico. En la Tabla 3.6 se compara el desempeño de nuevo protocolo MAC (ECRDA) versus el desempeño de CRDA, en términos del PRA. El escenario a comparar se compone de 140 terminales distribuidos de la siguiente manera: 100 de tráfico de voz, 5 de tráfico conversacional, 5 de tráfico interactivo, 5 de tráfico unidireccional y 25 de tráfico de fondo.

Tabla 3.6 ECRDA vs CRDA / Métrica de comparación - Promedio por Retardo de Acceso (PRA)

MAC	PRA VOZ (seg)	PRA CONVERSACIONAL (seg)	PRA INTERACTIVO (seg)	PRA UNIDIRECCIONAL (seg)	PRA FONDO (seg)
Umbral [19]	0.010	<< 1	~ 1	< 10	> 10
CRDA Prueba 1018	0.00448433	0.39687436	0.77915371	59.94646301	6.75392313
ECRDA Prueba 1021	0.00448433	0.02785462	0.02699183	1.02079610	81.96180000

El análisis de los datos obtenidos nos indica que el protocolo MAC (ECRDA) desacopla eficientemente los diferentes tipos de tráfico otorgándoles prioridad a aquellos que son más sensibles al retardo. De esta manera se observa que el valor de PRA en ECRDA se mantiene por debajo del umbral en todos los casos a excepción del PRA de fondo que se dispara a casi 82 segundos. En el caso de CRDA vemos que se comporta con justicia otorgándole a todos los tráficos la misma prioridad e identificamos que esa justicia impacta significativamente los valores de PRA para el caso conversacional y unidireccional. Con respecto a los terminales de voz, los valores de PRA son idénticos ya que la máxima prioridad de ambos protocolos es la voz.

En la Tabla 3.7 se compara ECRDA versus CRDA en función de los valores pico por retardo de acceso, ello con el objeto de darnos una idea de los valores máximos alcanzados. Para el caso de CRDA, el valor Pico sobrepasa los valores referenciales en todos los casos, excepto para la clase de fondo; a diferencia ECRDA mantiene los valores por debajo del umbral de referencia.

Tabla 3.7 ECRDA vs CRDA / Métrica de comparación – Valor Pico por Retardo de Acceso

MAC	PICO CONVERSACIONAL (seg)	PICO INTERACTIVO (seg)	PICO UNIDIRECCIONAL (seg)	PICO FONDO (seg)
Umbral Referencial [19]	<< 1	~ 1	< 10	> 10
CRDA Prueba 1018	1.91303962	7.02496390	147.65103734	38.58589180
ECRDA Prueba 1021	0.07475692	0.11410281	6.74283254	189.40797574

Habiendo comprobado que el protocolo MAC (ECRDA) de esta tesis, se desempeña eficientemente y mejor que [2], se procedió a evaluar la capacidad de atención de ECRDA en función de la cantidad de terminales de datos. Para ello se

han realizado comparaciones variando la cantidad de terminales de datos: Pruebas 1020, 1021 y 1022.

La Tabla 3.8 resume los resultados obtenidos. Se observa que el escenario de la Prueba 1020 sobrecarga el desempeño del protocolo. Se cumple con atender eficientemente a los tráficos de voz, conversacional e interactivo pero en desmedro del tráfico unidireccional y la indisponibilidad para el tráfico de fondo. La carga del sistema es tan alta que ocasiona que el tráfico de fondo nunca sea atendido. Manteniendo los 140 terminales, el escenario de la Prueba 1021, reduce la cantidad de terminales con prioridad y aumenta la cantidad de terminales de fondo a 25, mostrando un mejor desempeño y dentro de los umbrales permitidos.

Tabla 3.8 ECRDA / Métrica de comparación PRA – Capacidad para atender cierta cantidad de terminales móviles

Caso	PRA VOZ (seg)	PRA CONVERSACIONAL (seg)	PRA INTERACTIVO (seg)	PRA UNIDIRECCIONAL (seg)	PRA FONDO (seg)
Prueba 1020	0.00445597	0.02809235	0.05864087	64.68317789	sin datos
Prueba 1021	0.00447262	0.02722225	0.05579446	1.59422394	474.53193566
Prueba 1022	0.00446603	0.02704221	0.05555991	1.10048906	376.44013217

El escenario de la Prueba 1022 con 120 terminales, muestra también un buen desempeño y sus resultados difieren ligeramente del escenario de la Prueba 1021. Para amplificar dicha diferencia se procedió al análisis de valores pico, cuyos resultados se muestran en la Tabla 3.9. En esta tabla se observa que, para el escenario de la prueba 1021, el valor pico del tráfico unidireccional sobrepasa ligeramente el valor umbral permitido, lo cual es considerado todavía aceptable debido a que la diferencia no es significativa. Para el tráfico de fondo se detectan retardos alargados tanto en los escenarios de las pruebas 1021 y 1022. Por lo tanto se infiere que un escenario seguro de buen desempeño del protocolo MAC

eficiente (ECRDA) es el de la prueba 1022 (con 120 terminales); mientras que el de la prueba 1021 (con 140 terminales) se considera un escenario con un desempeño aceptable. Una anotación adicional es que los escenarios pueden variar pero en todos ellos se notará que nuestro protocolo MAC eficiente atenderá prioritariamente el tráfico más sensible.

Tabla 3.9 ECRDA / Métrica de comparación Valor Pico por Retardo de Acceso - Capacidad para atender cierta cantidad de terminales móviles

Caaso	PICO CONVERSACIONAL (seg)	PICO INTERACTIVO (seg)	PICO UNIDIRECCIONAL (seg)	PICO FONDO (seg)
Prueba 1020	0.07910716	0.49574129	161.07190894	sin datos
Prueba 1021	0.08710761	0.19729782	10.81127594	893.49575982
Prueba 1022	0.09992410	0.13680913	6.98515201	901.19534077

CAPÍTULO 4

CONCLUSIONES Y FUTUROS TRABAJOS

4.1 Conclusiones

Las conclusiones que se obtienen a la culminación de esta tesis son:

1. Se ha desarrollado un protocolo MAC eficiente para redes inalámbricas celulares de 3G, 3G+, usando el simulador de eventos discretos OMNET++. La eficiencia del protocolo MAC se logra programando una política de priorización dentro del algoritmo de planificación (scheduling); esto permite otorgar calidad de servicio a las aplicaciones que lo requieran (capítulo 3, sección 3.2.3).
2. Se ha diseñado el protocolo MAC de esta tesis tomando como base el protocolo CRDA; por lo que ha sido importante inferir el funcionamiento de este último usando técnicas de modelamiento como diagramas de flujo, diagramas de señalización y fórmulas para diseñar la composición de la trama TDMA. Luego, el diseño inferido ha sido portado a OMNET++ y validado mediante simulación (capítulo 3, sección 3.2.1).
3. Las simulaciones realizadas para lograr los resultados de esta tesis tuvieron tres objetivos generales: el primero fue validar el protocolo CRDA programado en OMNET++, el segundo fue probar el desempeño de CRDA con nuevas aplicaciones que demandaban distintos requerimientos de QoS y el tercero fue probar el desempeño del protocolo MAC de esta tesis (llamado ECRDA) y compararlo con CRDA en términos que demuestren el mejor desempeño en cuanto a QoS (capítulo 3 y anexo E, sección E.2).
4. OMNET++ ha servido para programar y validar el protocolo CRDA, obteniendo resultados similares a [2], con la misma tendencia estadística y comprobando que OMNET++ es una herramienta de software libre confiable para la simulación de eventos discretos.

5. Se comprueba que los protocolos MAC que aplican “igualdad” en el reparto de recursos, como [2], no otorgan el mejor desempeño cuando se enfrentan a aplicaciones que demandan diferentes requisitos en cuanto a QoS, por ejemplo: ancho de banda y demora en la atención. Esta situación ha sido demostrada (capítulo 3, sección 3.2.2) en donde se observa que en los escenarios probados se sobrepasan los umbrales permitidos para las aplicaciones, lo cual obliga a reducir la capacidad de atención del protocolo mediante la disminución de terminales de datos. De esta manera, el protocolo MAC de esta tesis, demuestra ser más eficiente al administrar los diferentes flujos de tráfico ofreciendo una mayor capacidad de atención (capítulo 3, sección 3.2.3) que el protocolo sin soporte de QoS.
6. La capacidad de atención en términos de cantidad de terminales móviles puede variar dependiendo de la configuración o escenario de prueba. Para nuestro caso consideramos 100 terminales de voz y luego se fue variando la cantidad de terminales de datos. Con 20 terminales de datos (prueba 1022) se logra un desempeño óptimo y seguro; sin embargo con 40 terminales de datos (prueba 1021) se logran aún resultados aceptables. Los escenarios de prueba 1021 y 1022 difieren en la cantidad de terminales con tráfico de fondo. Más allá de 25 terminales (con tráfico de fondo), el valor por retardo de acceso crece significativamente afectando la capacidad de atención de protocolo MAC (capítulo 3, sección 3.2.3).

4.2 Futuros trabajos de investigación

Para futuros trabajos de investigación se dejan los siguientes comentarios:

1. En este trabajo de tesis, se ha desarrollado un protocolo MAC eficiente orientado en un entorno ad-hoc y usando la herramienta de software libre OMNET++. El protocolo MAC eficiente puede ser utilizado como un prototipo que permita simular otros escenarios para la resolución de problemas relacionadas con las redes de comunicaciones inalámbricas

celulares. En ese sentido es posible realizar investigaciones sobre transmisión de tráfico IP sobre el protocolo MAC eficiente para analizar el comportamiento del mismo, por ejemplo en la presencia de retardo en la respuesta, para alguna situación o problemática en particular.

2. Se recomienda ampliar la investigación en el uso de técnicas de múltiple portadora (MIMO, múltiple entrada y múltiple salida). Para ello el framework del protocolo MAC eficiente debe ser re-editado para soportar interfaces físicas.
3. Se puede re-editar el framework del protocolo MAC eficiente para el caso puntual de codificar los canales de información (i-slots) usando CDMA y cuya finalidad sería ampliar la capacidad de multiplexación del protocolo MAC para soporte de mayor tráfico de subida y por ende mayor capacidad de terminales móviles o servicios que demandan gran ancho de banda.
4. Se sugiere probar otras técnicas de doblado como TDD para las tramas de subida y bajada.

BIBLIOGRAFÍA

- [1] D J Goodman, R A Valenzuela, K T Gayliard, and B Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications", *IEEE Trans. Commun.*, vol. 37, pp. 885-890, Aug. 1989.
- [2] Luciano Lenzini, Marco Luise and Ruggero Reggiannini, "CRDA: A Collision Resolution and Dynamic Allocation MAC Protocol to Integrate Data and Voice in Wireless Networks ", *IEEE Journal On Selected Areas in Commun.*, vol. 19, NO. 6, pp. 1153-1163, Jun. 2001.
- [3] András Varga, "OMNeT++: Discrete Event Simulation System, version 3.2, User Manual", March. 2005.
- [4] UIT-R Recomendación M.1035, "Marco General para el Estudio de la Funcionalidad de las Interfaces Radioeléctricas y del Subsistema Radioeléctrico en las Telecomunicaciones Móviles Internacionales-2000 (IMT-2000)"
- [5] REDES DE COMPUTADORAS, Andrew S. Tanenbaum (1999), Tercera Edición, Cap 4 : La Subcapa de Acceso al Medio, páginas 243-246.
- [6] A GUIDE TO THE WIRELESS ENGINEERING BODY OF KNOWLEDGE (WEBOK), G. Giannattasio, J. Erfanian, P. Wills, H. Nguyen, T. Croda, K. Rauscher, X. Fernando, N. Pavlidou, and K. D. Wong (2009), IEEE PRESS, Primera Edición, Chapter 1 : Wireless Access Technologies, pages 4-7
- [7] I. F. Akyildiz, J. McNair, L. Carrasco, R. Puigjaner, Y. Yesha, "Medium Access Control Protocols for Multimedia Traffic in Wireless Networks ", *IEEE Network*, July 1999.
- [8] Ralph. Santitoro, "Introduction to Quality of Service (QoS)", Nortel Networks, 2005.
- [9] J. De Vile, "A reservation multiple access scheme for an adaptive TDMA air interface", in *Proc. 4th WINLAB Workshop Third Generation Wireless Information Networks.*, NJ, Oct. 1993.
- [10] G. Anastasi, D. Grillo, L. Lenzini, and E. Mingozzi, "A bandwidth reservation protocol for speech/data integration in TDMA-based advanced mobile systems", in *Proc. INFOCOM '96*, San Francisco, CA.
- [11] P. T. Brady, "A model for generating ON-OFF speech patterns in two-way conversations ", *Bell Syst. Tech. J.*, pp. 2445-2472, Sept. 1969.

- [12] J. A. Moreno, J. Padilla, V. Escobar, A. Correo, "Caracterización y simulación del tráfico de redes LAN mediante el modelo MMPP ", Rev. Fac. Univ. Antioquia. N.º42., pp. 7-29, Dic. 2007.
- [13] W. E. Leland, M.S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic", IEEE/ACM Trans. Networking, vol. 2, pp. 1-15, Feb. 1994.
- [14] R. Vecchio, "Modelización de tráfico autosimilar y evaluación de sus efectos en el tamaño de las colas", Instituto Tecnológico de Buenos Aires, www.itba.edu.ar/capis/epg-tesis-y-tf/vecchio-tesisdemagister.pdf., 2003.
- [15] H. Heffes, and D. M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance", IEEE J. Select. Areas. Commun, vol. SAC-4, Sept. 1986.
- [16] S. Deng, "Empirical Model of WWW Document Arrivals at Access Link", Proceeding of ICC, 1996.
- [17] UIT-R Recomendación M.1034, "Requisitos de las interfaces radioeléctricas para los futuros sistemas públicos de telecomunicaciones móviles terrestres (FSPTMT)", 1994
- [18] ITU-R Recommendation M.1645, "Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000", 2003
- [19] UIT-R Recomendación M.1079-2, "Requisitos relativos a la calidad de funcionamiento y servicio en las redes de acceso a las telecomunicaciones móviles internacionales-2000 (IMT-2000)", 2003
- [20] The 3rd Generation Partnership Project, Tech. Rep. TR25.896 V6.6.0, "Feasibility Study for enhanced uplink for UTRA FDD", March 2004
- [21] A. Urie, M. L. Streeton, and C. Mouro, "An advanced TDMA mobile access system for UMTS", *IEEE Personal Commun.*, vol. 2, pp. 38-47, Feb. 1995.
- [22] A. Klemm, C. Lindemann, and M. Lohmann, "Traffic Modeling and Characterization for UMTS Networks", *IEEE*, 2001

ANEXOS

ANEXO A

Protocolos MAC para redes inalámbricas

Un protocolo de control de acceso al medio (MAC) con técnica de doblado FDD, para una red inalámbrica de paquetes, hace posible que varios terminales móviles puedan transmitir tráfico de diferentes velocidades (por paquetes) hacia una estación base (enlace de subida) dentro de un medio compartido.

A continuación se describe en orden evolutivo algunos protocolos MAC para redes inalámbricas de paquetes que usan técnica dúplex FDD, dando énfasis en el enlace de subida (desde el terminal móvil hacia la estación base):

A.1 Listado de Protocolos

A.1.3 PRMA: Acceso Múltiple por Reserva de Paquete [1]

PRMA ha sido desarrollado como un medio eficiente que maneja servicios múltiples sobre un enlace inalámbrico.

PRMA combina las mejores características de los protocolos de circuito conmutado y paquete conmutado

Alguna de las características son las siguientes:

- Disputa equitativa por el acceso al recurso (canal) y basado en la probabilidad de permiso, que es un parámetro de diseño y estabilidad del sistema.
- Utiliza VAD para mejorar la capacidad y el desempeño del sistema.
- Usa una trama TDMA para dividir el canal en ranuras de tiempo (*time-slot*), los cuales son agrupados en tramas (*frames*) de tamaño fijo.
- La duración de la trama ha sido seleccionado de tal manera que los terminales de voz transmiten un simple paquete por trama.
- Las ranuras de tiempo están clasificados como “disponible” o “reservado”.
- Los terminales móviles compiten por los slots marcados como “disponibles” usando el protocolo ALOHA Ranurado. Luego de tener éxito en la disputa por la ranura de tiempo (*time-slot*), el terminal reserva el slot por el tiempo

que tome los datos (voz ó datos) que va a transmitir. Los otros terminales están prohibidos de transmitir durante el tiempo de reserva del slot. Luego de que el terminal móvil no tiene más datos por enviar, la ranura de tiempo se libera y se marca como “disponible” para que sea disputado por otros terminales móviles ansiosos por transmitir.

- Se asigna baja probabilidad de permiso a la transmisión de datos y alta probabilidad de permiso a la transmisión de voz.

A.1.4 PRMA++: Acceso Múltiple por Reserva de Paquete Mejorado [9]

PRMA++ es una versión mejorada del protocolo PRMA desarrollado por Goodman [1]. El protocolo se esquematiza en las figuras A.1 y A.2.

Alguna de las características mas relevantes de este protocolo son las siguientes:

- Este protocolo MAC para redes inalámbricas separa usuarios y servicios en diferentes ranuras de tiempo.
- PRMA++ divide la trama TDMA en ranuras de tiempo (time-slots) marcados como ranuras de reservación (R-slot), ranuras de información (I-slot) y ranuras disponibles (A-slot)
- En el enlace de subida (estación móvil a estación base), las ranuras R-slot son usados solamente como slots de reserva. El tráfico de voz y datos se transmite en las ranuras I-slots.
- Cada ranura R-slot en el enlace de subida, tiene asociada una ranura A-slot en el enlace de bajada (estación base a estación móvil); esta última se utiliza para el reconocimiento de solicitudes de canal. Igualmente el enlace de bajada cuenta también con ranuras I-slots que transportan el tráfico de bajada para voz y datos.
- Una estación móvil que desea transmitir datos sobre el canal primero tiene que enviar la solicitud de reserva en el time-slot marcado como ranura de reservación. (R.slot) usando acceso aleatorio.
- Cuando dos o más estaciones tratan de transmitir una solicitud de canal al mismo instante se genera una colisión. El mecanismo de competencia

(disputa) por el canal usado para resolver las colisiones, permite que las estaciones móviles retransmitan sus solicitudes en el siguiente slot de reserva y con una probabilidad de reintento (parámetro de diseño) fija.

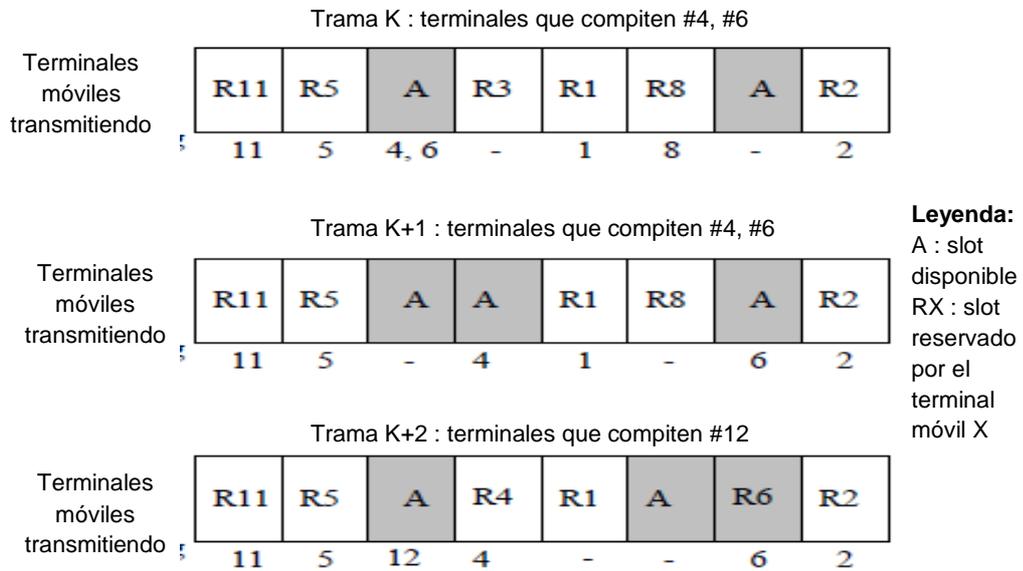


Figura A.1 PRMA, Fuente [1]

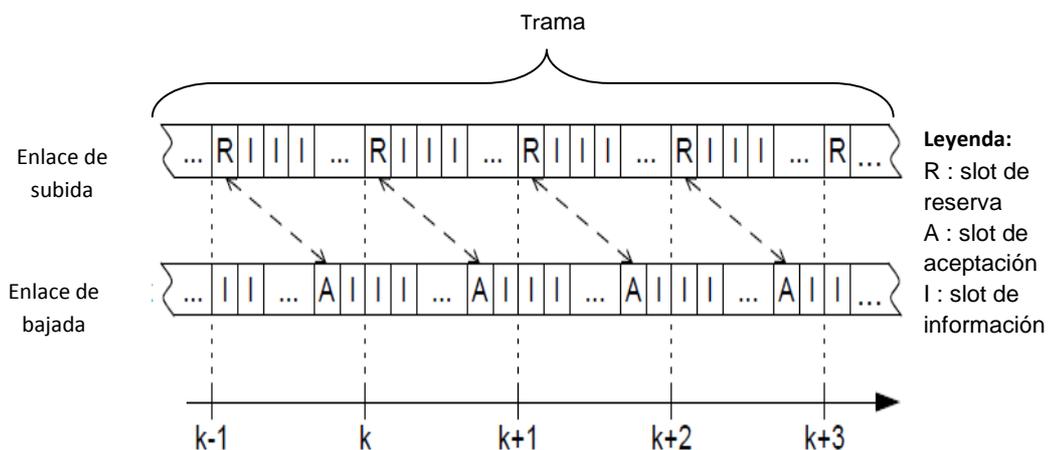


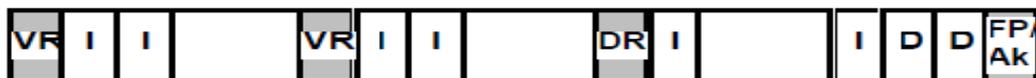
Figura A.2 PRMA++, Fuente [9]

A.1.5 SIR: Acceso por Radio de Servicios Integrados [10]

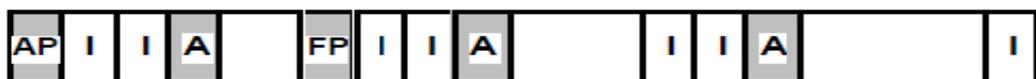
Este protocolo está basado en PRMA++ y puede acomodar requerimientos de voz y datos por ráfagas de una manera eficiente. La figura A.3 esquematiza la estructura interna de este protocolo.

Algunas características de este protocolo son las siguientes:

- Es un protocolo libre de contención en la medida que separa las ranuras de reserva en voz y datos
- Utiliza el ancho de banda de una manera flexible de tal forma que acomoda la demanda de tráfico de datos a la vez que asegura el ancho de banda garantizado para aplicaciones de voz.
- Este protocolo maneja el tráfico de voz de la misma forma que PRMA++.
- En relación al tráfico de datos introduce una mejora en su algoritmo de asignación de recursos, el cual consiste en realizar encuestas (*en inglés polling*) a las estaciones móviles, cada período de tiempo determinado y con la finalidad de recoger las solicitudes de transmisión de las fuentes de datos.
- En cuanto al esquema de la trama de subida, separa los canales de reserva e información en solo voz y solo datos. De esta forma se mantienen canales independientes para ambos servicios, garantizando que el tráfico de voz y datos tengan canales disponibles todo el tiempo.



ENLACE DE SUBIDA



ENLACE DE BAJADA

Leyenda:

VR slot de reserva para voz

I : slot de información

A : slot de aceptación

FP.AK. slot de notificación de aceptación

DR : slot de reserva para datos

AP : patrón de asignación

FP : slot de notificación

Figura A.3 SIR, Fuente [10]

A.1.6 CRDA: Protocolo MAC para integración de voz y datos en redes inalámbricas con resolución de colisiones y asignación dinámica [2]

Este protocolo usa la técnica de acceso múltiple por reserva de paquete ranurado con canales de reserva dedicados y codificados con CDMA, lo cual permite superar el principal defecto de los esquemas PRMA existentes: la contención (disputa) por el recurso. La figura A.4 esquematiza este protocolo.

Las características de este protocolo son las siguientes:

- Elimina las colisiones que se originan por la concurrencia de solicitudes de reserva de ancho de banda realizadas por dos o más estaciones móviles dentro del mismo slot de reserva.
- La resolución de las colisiones tiene lugar directamente en la capa física al introducir el modo de transmisión S-CDMA en los slots de reserva.
- En el enlace de subida, las estaciones móviles notifican a la estación base el tipo de servicio que soportarán y los requerimientos de recursos (canales) para transmitir el tráfico deseado.
- La trama del enlace de subida está constituida por slots de reserva (R-slot) y slots de información (I-slot)
- Las reservas de ancho de banda se realizan por los R-slot, mientras que los I-slot llevan el tráfico de voz y datos. De esta manera las solicitudes de reserva no se bloquean cuando todos los I-slot están ocupados con tráfico.
- La estación base mantiene control centralizado sobre la política de asignación de los I-slot (en inglés scheduling algorithm).
- La trama del canal de bajada está constituido por slots de reconocimiento (en inglés acknowledgement) llamados A-slot y slots de información, I-slot.
- El uso de códigos CDMA en el enlace de subida, permite que la estación base sepa cuantos y que usuarios están solicitando ancho de banda. De esta forma se tiene un esquema de reconocimiento de paquete que supera el problema de colisión en el R-slot, mejorando el desempeño del sistema.

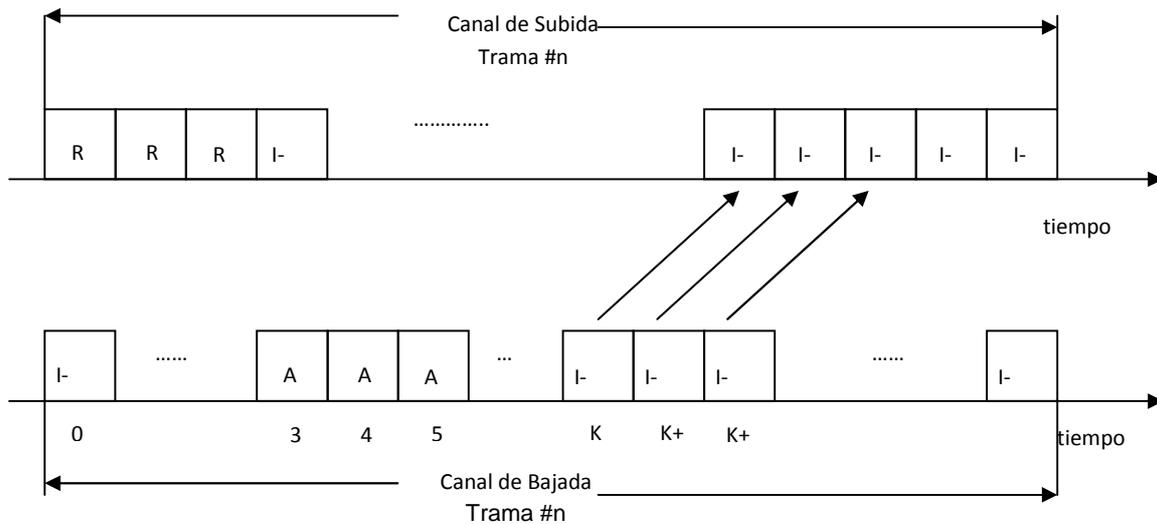


Figura A.4 CRDA, Fuente [2]

A.2 Ventajas y Desventajas de los protocolos MAC mencionados

A continuación se exponen los argumentos que diferencian cada uno de los protocolos MAC mencionados anteriormente.

PRMA

Ventajas:

- Explota el patrón característico de la conversación que se compone en períodos de conversación y silencio (ON-OFF); ello permite maximizar la comunicación al enviar solo aquellos paquetes que contienen períodos de conversación digitalizados. Para los períodos de silencio se envían paquetes tontos o vacíos (*en inglés dummy*).
- Con la finalidad de transmitir paquetes, los cuales tienen un comportamiento por ráfagas, PRMA hace uso del protocolo de acceso aleatorio ALOHA Ranurado.

Desventajas:

- Explota sólo el 37% de la capacidad del canal.
- Se requiere realizar la elección adecuada del parámetro de probabilidad de permiso una vez ocurrida la colisión

- El protocolo es sensible a las colisiones
- Todas las ranuras de tiempo (slots) de una trama (frame) de subida, se emplean para solicitar el envío de tráfico o para transmitir tráfico, entonces al no existir diferenciación en las ranuras (slots) de la trama de subida se puede producir alta competencia de los móviles por la toma de una ranura, disminuyendo la posibilidad de asignación de slots para la transmisión de tráfico lo cual afectaría el desempeño (*throughput*) del protocolo.

PRMA++

Ventajas:

- Clasifica las ranuras de tiempo en slots de reserva (R-slot), slots de reconocimiento (A-slot) y slots de información (I-slot), tanto para la trama de subida como de bajada
- Asigna slots dedicados a la contención y al tráfico

Desventajas:

- Al descender de PRMA adolece del problema de estabilidad ocasionado por el mal diseño del parámetro de probabilidad de permiso que no es otra cosa que el mecanismo de resolución de colisiones que usa el protocolo de las estaciones móviles para retransmitir sus solicitudes en los siguientes slots de reserva cada vez que el paquete de solicitud se perdió debido a una colisión.
- El protocolo no ha sido diseñado para transmitir tráfico de datos

SIR

Ventajas:

- Para probar el desempeño del protocolo los autores modelaron el tráfico de datos basándose en la Distribución Pareto que es más apropiada para aplicaciones en redes de datos que la distribución Poissoniana usualmente usada para tráfico de voz
- Minimiza las colisiones al desdoblarse la trama TDMA en slots de reserva e información para voz y datos

- Protocolo orientado a la transmisión de datos

Desventajas:

- Al utilizar ranuras (slots) dedicados para voz y para datos disminuye la capacidad de ranuras de información usadas para transmitir tráfico

CRDA

Ventajas:

- Supera el problema de las colisiones en las ranuras de reserva al introducir codificación S-CDMA en el R-slot
- Debido a lo anterior, las solicitudes de reserva no se bloquean cuando todos los I-slots de subida están ocupados
- Durante la fase de reserva se realiza la diferenciación del servicio, es decir que las estaciones móviles informan a la estación base acerca del tipo de tráfico que desean transmitir.
- El protocolo tiene soporte para la transmisión de tráfico de datos con longitud variable de paquete, con ello se evita la sobrecarga en las solicitudes de reserva de canal.
- Durante la fase de asignación de recurso, la estación base maneja dos tipo de colas: una cola FIFO (Primero en entrar, primero en salir) para voz, y una cola de Círculo Aleatorio (Round Robin) para datos.

Desventajas:

- Las colas de Círculo Aleatorio (Round Robin) usadas para datos son atendidas de una manera circular, es decir no usan priorización para las colas de datos
- El protocolo no tiene soporte para fuentes de tráfico de velocidad variable

ANEXO B

Modelamiento de fuentes de tráfico para datos

B.1 Características de las fuentes de tráfico de datos

B.1.1 Definición de tráfico auto similar

Como indica [13], el término “auto similar” (en inglés self-similar) fue acuñado por Mandelbrot, este último y sus colegas captaron la atención de los investigadores en estadística, al definir los procesos auto similares aplicados principalmente en áreas como hidrología y geofísica. La auto similitud ocurre en fenómenos geométricos naturales. Comúnmente, este tipo de propiedad es también llamado fractal.

Un proceso se define como auto similar cuando este es similar en diferentes escalas de espacio ó tiempo. En procesos aleatorios de tráfico de datos, el fenómeno de auto similitud ocurre en la dimensión del tiempo. Esto tiene consecuencias significantes en la estadística del tráfico.

A continuación se explica la definición matemática para un proceso auto similar como el expuesto en [13]:

Sea $\mathbf{X} = (X_t : t = 0, 1, 2\dots)$ un proceso estocástico estacionario en co-varianza con media μ , varianza σ^2 y función de autocorrelación $r(k)$, $k \geq 0$. En particular, se asume que \mathbf{X} tiene una función de autocorrelación de la forma:

$$r(k) \sim k^{-\beta} L(k) \quad (\text{B.1})$$

Donde:

el valor de k tiende al infinito positivo,

$$0 < \beta < 1$$

el valor de L varía lentamente hacia el infinito

Por simplicidad se asume a L constantemente asintótico.

Para cada $m = 1, 2, 3, \dots$ sea $\mathbf{X}^{(m)} = (X_k^{(m)} : k=1, 2, 3, \dots)$ una nueva serie de tiempo de co-varianza estacionaria con correspondiente función de autocorrelación $r^{(m)}$, obtenida al promediar la serie original \mathbf{X} sobre bloques de tamaño m .

El proceso \mathbf{X} es llamado exactamente auto similar (de segundo orden) y con parámetro de auto similitud:

$$H = 1 - \frac{\beta}{2} \quad (\text{B.2})$$

Sí para todo $m = 1, 2, \dots$

$$r^{(m)}(k) = r(k), k \geq 0 \quad (\text{B.3})$$

En otras palabras, \mathbf{X} es exactamente ó asintoticamente auto similar si los correspondientes procesos agregados $\mathbf{X}^{(m)}$ son los mismos que \mathbf{X} o llegan a ser indistinto de \mathbf{X} ; al menos en sus funciones de autocorrelación.

Así mismo \mathbf{X} es llamado asintoticamente auto similar (de segundo orden), con parámetro de auto similitud

$$H = 1 - \frac{\beta}{2} \quad (\text{B.4})$$

Sí $r^{(m)}(k) \rightarrow r(k)$, para todo k y m con valores grandes (B.5)

El parámetro de auto similitud lo determina H (parámetro de Hurst) y el valor de β varía entre 0 y 1.

Matemáticamente la auto similitud se manifiesta en diferentes maneras:

- La varianza de la media de la muestra decrece más lentamente que la recíproca del tamaño de la muestra (varianza que decae lentamente)
- La autocorrelación decae hiperbólicamente y no exponencialmente rápida, lo cual implica una función de autocorrelación no aditiva (dependencia de largo rango)
- La densidad espectral $f(\cdot)$ obedece una ley de potencia (en inglés power-law) cerca del origen

La característica más resaltante de ser un proceso (exactamente ó asintóticamente) auto similar de segundo orden, es que los procesos agregados $X^{(m)}$ poseen una estructura de correlación no degenerativa tal como m va tendiendo a infinito. Esta conclusión difiere de las proposiciones típicas para modelado de tráfico por paquetes, los cuales sostienen que los procesos agregados tienden a ruido puro en el segundo orden, es decir la correlación $r^{(m)}(k)$ tiende a cero conforme m tiende a infinito. La característica no degenerativa de un proceso auto similar se compara gráficamente con la de un proceso no auto similar en la figura B.1.

B.1.2 Dependencia de Largo Plazo

Como indica [14], una de las propiedades más significantes de los procesos auto-similares está referida a la dependencia de largo plazo. Esta propiedad está definida en términos del comportamiento de la autocovarianza $C(\tau)$ a medida que τ se incrementa.

Para muchos procesos, la autocovarianza decae rápidamente con τ . Por ejemplo, para procesos de incrementos Poisson con incrementos L y media λ , la autocovarianza para valores de $\tau > L$ es:

$$C(\tau) = R(\tau) - \lambda^2 = \lambda^2 - \lambda^2 = 0 \quad (\text{B.6})$$

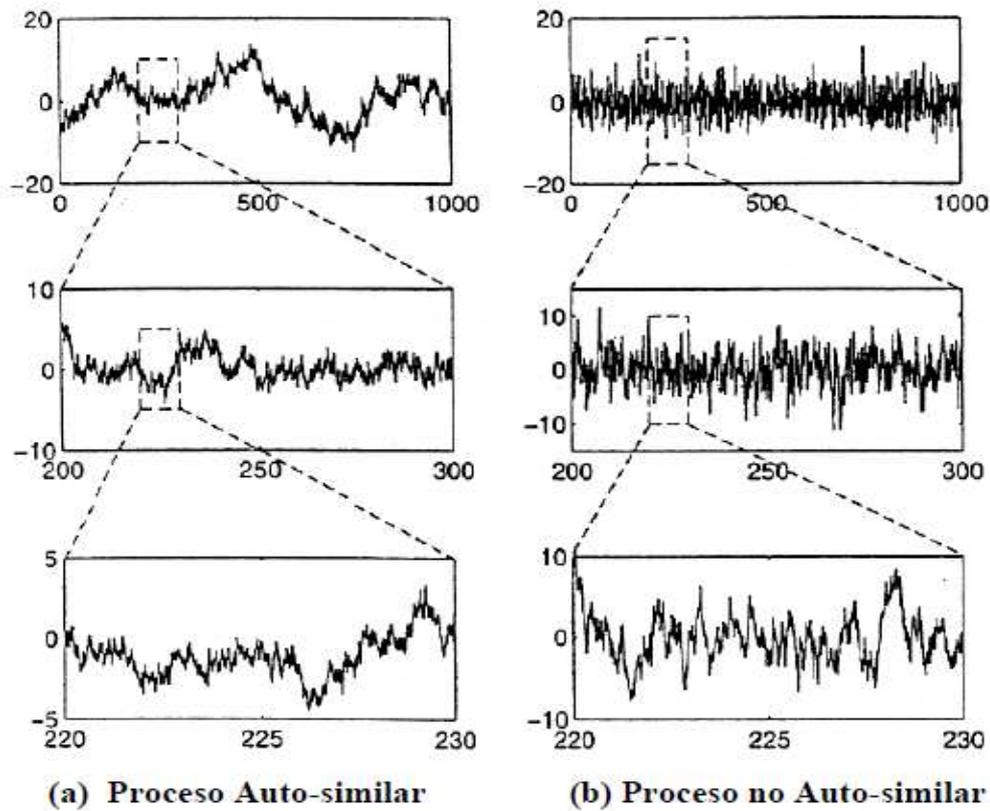


Figura B.1 Trafico Auto similar, Fuente [14]

Un proceso dependiente de corto plazo satisface la condición de que su autocovarianza decae al menos tan rápidamente como una exponencial:

$$C(k) \approx a^{|k|}, \text{ a medida que } |k| \rightarrow \infty, 0 < a < 1, \quad \text{(B.7)}$$

Donde \approx denota que las expresiones a ambos lados son asintóticamente proporcionales una a otra.

En contraste, un proceso dependiente de largo plazo tiene una covarianza que decae hiperbólicamente:

$$C(k) \approx |k|^{-\beta}, \text{ a medida que } |k| \rightarrow \infty, 0 < \beta < 1, \quad (\text{B.8})$$

Donde β es el mismo parámetro definido anteriormente y está relacionado al parámetro de Hurst mediante $H = 1 - \beta/2$. En este caso $\sum_k C(k) \rightarrow \infty$.

La dependencia de largo plazo refleja intuitivamente el fenómeno de persistencia en un proceso auto-similar, además de la existencia de ráfagas (*en inglés burstiness*) en todas las escalas de tiempo.

B.1.3 Densidad Espectral

También se pueden dar expresiones equivalentes de la dependencia de largo plazo en el dominio de la frecuencia. Específicamente, la densidad espectral de potencia obedece a una ley de potencia cerca del origen:

$$S(\omega) \approx \frac{1}{|\omega|^\gamma}; \text{ si } \omega \rightarrow 0 \text{ para } 0 < \gamma < 1 \quad (\text{B.9})$$

La densidad espectral para un proceso estocástico discreto en el tiempo se define de la manera siguiente:

$$S(\omega) = \sum_{-\infty}^{+\infty} R(k) e^{-j2k\omega} \quad (\text{B.10})$$

$$S(0) = \sum_{-\infty}^{+\infty} R(k) \quad (\text{B.11})$$

Puede demostrarse que:

$$\gamma = 1 - \beta = 2H - 1 \quad (\text{B.12})$$

En contraste, los procesos dependientes de corto plazo están caracterizados por una densidad espectral que permanece finita mientras $\omega \rightarrow 0$. Esto ocurre cuando $\gamma = 0$ ó equivalentemente, $H = 0.5$.

En términos de la función de autocorrelación, resulta un valor infinito de $S(0)$ si los valores de $R(k)$ no decaen lo suficientemente rápido para valores grandes de k , que conforman una suma finita. Las expresiones de $C(k)$ y de $S(\omega)$ pueden ser útiles en el chequeo de auto similitud.

B.1.4 Distribuciones de cola pesada

Otra forma de definir los procesos estocásticos autosimilares es mediante las distribuciones de cola pesada (en inglés heavy-tailed). Las distribuciones de cola pesada pueden ser usadas para caracterizar densidades de probabilidad que describen procesos de tráfico tales como tiempos de entre arribo de paquetes y duración de ráfagas.

Una variable aleatoria es de cola pesada si:

$$1 - F(x) = \Pr[X > x] \approx \frac{1}{x^\alpha} \text{ si } x \rightarrow \infty, \text{ para } \alpha > 0 \quad (\text{B.13})$$

Una variable aleatoria con una distribución de cola pesada muestra una varianza alta o infinita. Como ejemplo se tiene la distribución de Pareto. La figura B.2 compara las distribuciones de cola pesada tipo Pareto para distintos valores de α (alpha).

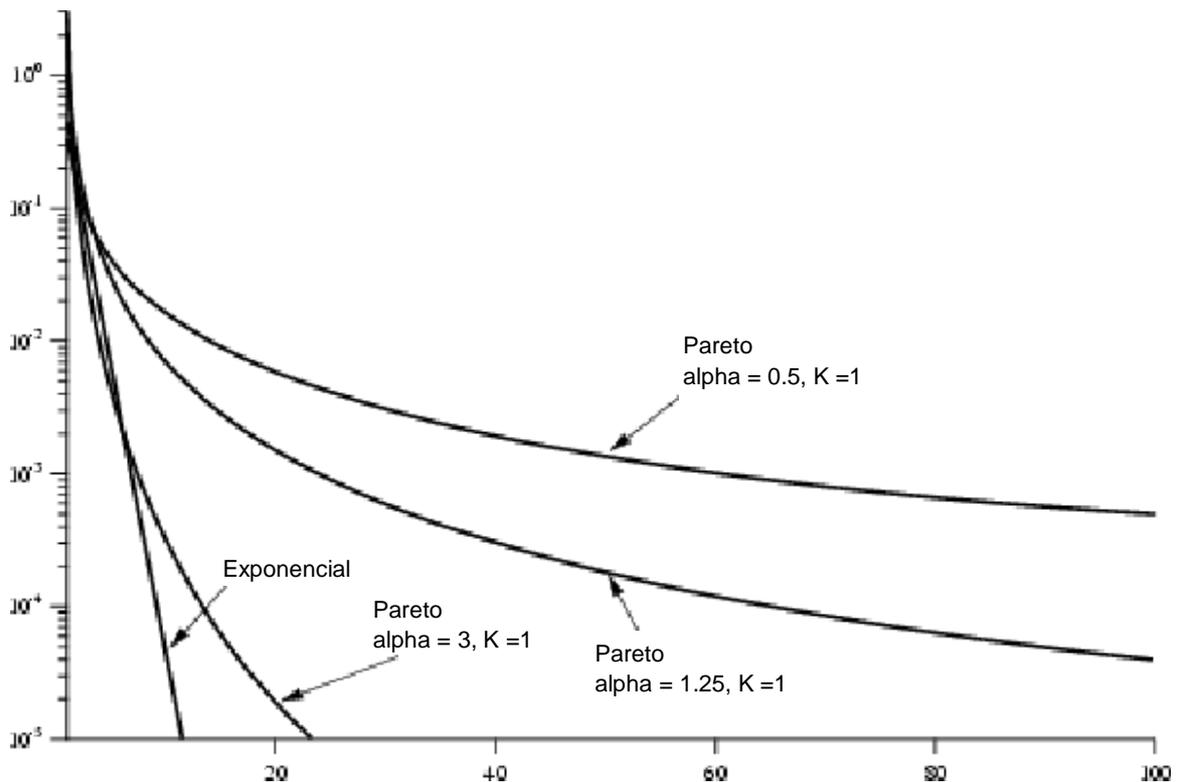


Figura B.2 Distribuciones de Cola Pesada: Pareto and Exponential Probability Density Functions, Fuente [14]

B.2 Metodología sugerida para el Modelado de Datos

Para [12], el diseño de un modelo exige dos etapas: primero el ajuste de cierto proceso estadístico, al que suele denominarse caracterización; segundo: seleccionar el modelo y ajustar el proceso estadístico en cuestión.

La caracterización (Figura B.3) supone analizar la realidad que se va a imitar (el tráfico) con el objeto de describirla mediante un conjunto reducido de variables (descriptor estadístico) que resumen y abstraen sus aspectos más importantes. Tras la definición de este descriptor, que, en ocasiones puede actuar directamente como un modelo en sí, la tarea de modelado se centra en elegir un proceso matemático que aproxime dicho descriptor, ya con independencia de la realidad física detallada que resume. El objetivo de esta fase, tras decidir qué modelo se emplea, es una parametrización adecuada del mismo que asegure un ajuste óptimo del descriptor.

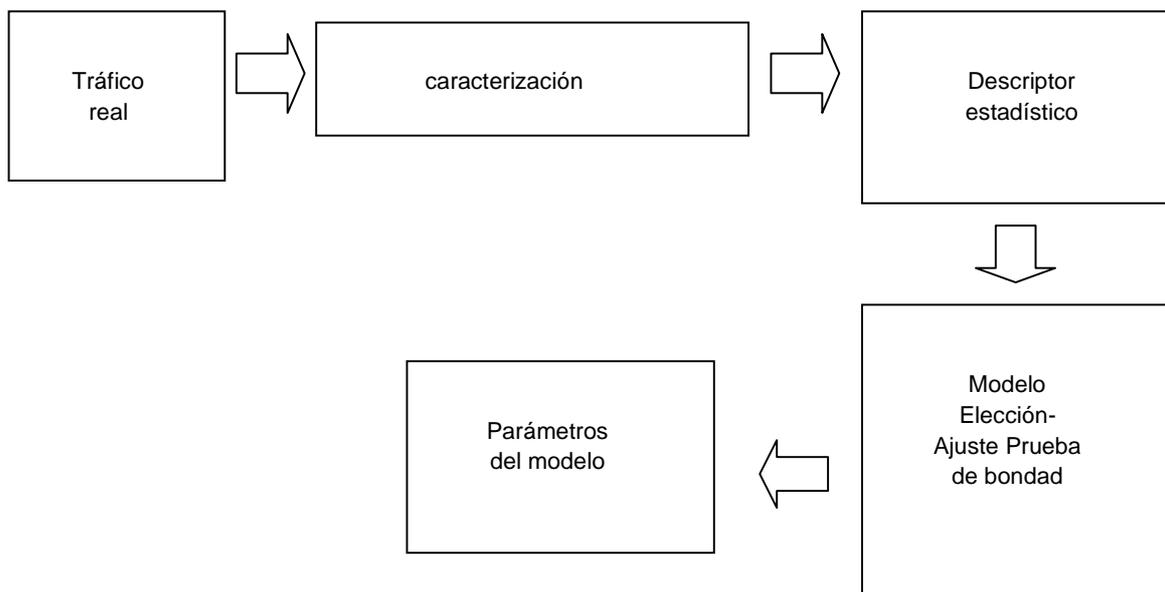


Figura B.3 Proceso de Diseño de un Modelo, Fuente [12]

Después de caracterizar el tráfico, debe realizarse el proceso de simulación (Figura B.4), que consiste en realizar una abstracción de un modelo real (ambiente real de

trazas de tráfico) y luego representarlo por medio de un modelo equivalente para obtener información relacionada con el modelo original. Esto con el fin de realizar pruebas que no sería posible realizar directamente sobre el modelo real de tráfico

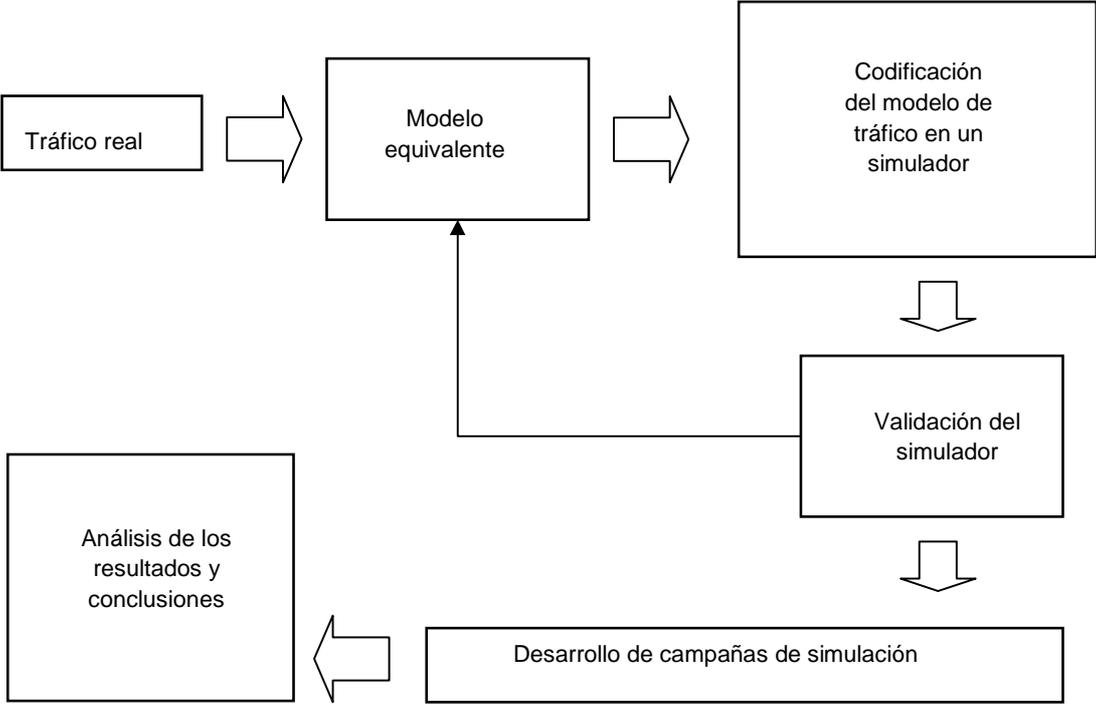


Figura B.4 Proceso de Simulación, Fuente [12]

ANEXO C

Calidad de Servicio (QoS)

La Calidad de Servicio (QoS) es un término amplio usado para describir la experiencia total que recibe de la red un usuario o aplicación. Hoy en día el tráfico de red es diverso y cada tráfico tiene sus propios requerimientos en términos de ancho de banda, retardo, pérdida y disponibilidad. La mayor parte de tráfico que se cursa en la red es IP. Las redes IP soportan distintas aplicaciones, muchas de ellas requieren menor latencia; de otra forma si la latencia es alta, se vería afectada significativamente la calidad percibida por el usuario final o las aplicaciones no funcionarían correctamente.

C.1 Medidas de desempeño para la QoS

Existe una serie de parámetros de QoS que ayudan a medir y monitorear si el nivel de servicio recibido u ofrecido se está alcanzando. De acuerdo a [8], estos parámetros son:

- La disponibilidad de red
- El ancho de banda
- El retardo (en inglés delay)
- La variación de retardo (en inglés jitter)
- La tasa por pérdida de paquetes - PLR (en inglés packet loss rate)/pérdida de bits – BER (en inglés bit error rate)

También existen otros parámetros que afectan el desempeño de la QoS, que no pueden ser medidos, pero proporcionan los mecanismos de administración para los elementos de la red:

- Prioridad de emisión de paquetes
- Prioridad de eliminación de paquetes

C.2 Dimensiones para el desempeño de la QoS por aplicación

La Tabla C.1 ilustra las consideraciones que se deben tener cuando se desea dimensionar el desempeño de las aplicaciones basado en un esquema de QoS. Si no se toman en cuenta dichas consideraciones el tráfico puede experimentar una conducta impredecible.

Tabla C.1 Dimensiones para el desempeño de la QoS por aplicación

Aplicación	Dimensiones para el Desempeño de la QoS			
	Ancho de Banda	Sensibilidad al		
		Retardo	Jitter	Pérdida de paquetes
VoIP	Baja	Alta	Alta	Alta
Video Conferencia	Alta	Alta	Alta	Media-Alta
Flujo de video por demanda	Alta	Media	Baja	Media-Alta
Flujo de Audio	Baja	Media	Baja	Media
Transacciones cliente/servidor	Media	Media	Baja	Alta
Correo electrónico	Baja	Baja	Baja	Alta
Transferencia de Archivos	Media	Baja	Baja	Alta

ANEXO D

Detalles de OMNET++

En este anexo se detallan los aspectos generales relativos al Framework de OMNET++. [3].

D.1 Jerarquía de los Módulos

Un modelo en OMNeT++ consiste de módulos jerárquicamente anidados que se comunican mediante el paso de mensajes entre ellos. Los modelos de OMNeT++ se llaman “networks”. El módulo superior es el “system module”, y el system module contiene “sub módulos” ; estos últimos pueden contener otros sub módulos (Figura D.1). La profundidad de los módulos no está limitada, esto permite que el usuario refleje la estructura lógica del sistema que desea modelar.

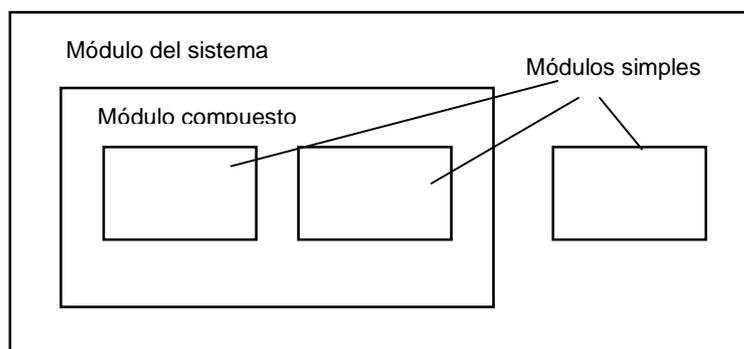


Figura D.1 Jerarquía de los Módulos en OMNET++, Fuente [3], OMNET++ Manual año 2005

La estructura del modelo se describe con el lenguaje NED de OMNeT++.

Los módulos que contienen sub-módulos se llaman “módulos compuestos” .

D.2 Mensajes, Puertas y Enlaces

Los módulos se comunican intercambiando mensajes entre ellos. Durante la simulación, los mensajes pueden representar tramas ó paquetes dentro de una red de computadoras, etc. Además los mensajes pueden contener estructuras de datos complejas. Los módulos pueden enviar mensajes directamente al destino o a través de una ruta predefinida por puertas y enlaces

Las puertas son las interfaces de entrada y salida del módulo, los mensajes se envían a través de puertas de “salida” y llegan por puertas de “entrada”.

Cada conexión ó enlace se crea dentro de cada nivel único de la jerarquía de módulos, dentro de un módulo compuesto (Figura D.2).

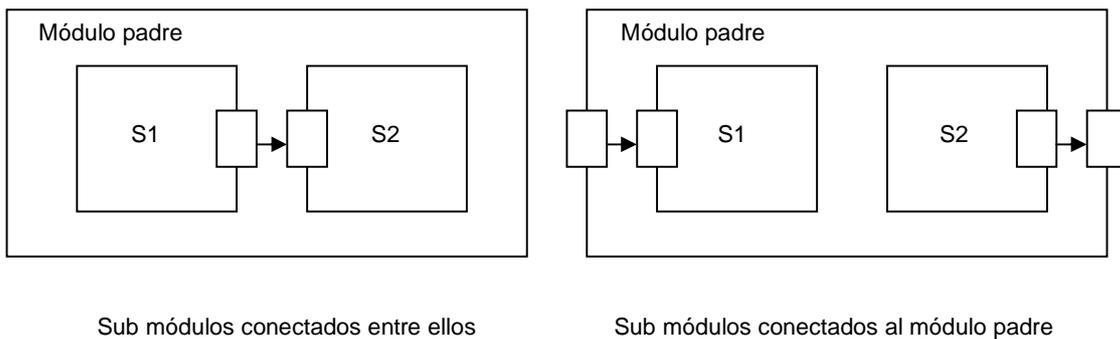


Figura D.2 Módulos y Sub Módulos conectados mediante puertas y enlaces,
Fuente [3], OMNET++ Manual año 2005

D.3 Parámetros

Los módulos pueden contener parámetros y los parámetros pueden ser asignados ya sea mediante los archivos NED o dentro del archivo de configuración “omnetpp.ini”. Los parámetros son usados para personalizar la conducta de los módulos simples y la topología del modelo.

D.4 Programación de los Algoritmos

Los módulos simples del modelo contienen algoritmos como funciones C++. De esta manera se puede explotar la flexibilidad y potencia del lenguaje de programación y soportado por la librería de clases del simulador OMNeT++.

Los objetos de simulación (mensajes, módulos, colas, etc) se representan mediante clases en C++. Ellos han sido diseñados para trabajar eficientemente, creando de esta manera una plataforma de programación y simulación potente.

Las siguientes clases son parte de la librería de clases de simulación:

- módulos, puertas y enlaces
- parámetros
- mensajes
- clases contenedoras
- clases para la colección de datos
- clases estadísticas y clases para la distribución de estimación (histogramas, etc)

D.5 La arquitectura de OMNET++

El simulador de eventos discretos OMNET++ tiene una arquitectura modular. El siguiente diagrama muestra la arquitectura de alto nivel de la simulación en OMNET++:

Los rectángulos de la figura D.3 representan los siguientes componentes:

- Sim, es el kernel de simulación y la librería de clases. Sim viene a ser una librería que se enlaza al programa de simulación.
- Envir, es otra librería que contiene todo el código que es común a todas las interfaces de usuario. Main() está contenido dentro de Envir. Además Envir proporciona los servicios para manejo de los archivos *.ini que ayudarán en implementaciones específicas para interfaces de usuario.
- Cmdenv y Tkenv vienen a ser implementaciones específicas para interfaces de usuario. Toda simulación se enlaza a uno de estos dos componentes.
- La Model Component Library comprende varias definiciones de módulos simples con su respectiva implementación en C++, tipos de módulos

compuestos, canales, networks, tipos de mensajes; en general todo lo que pertenece a modelos que han sido enlazados dentro del programa de simulación. El programa de simulación es capaz de ejecutar cualquier modelo que tenga todos los componentes necesarios enlazados a este.

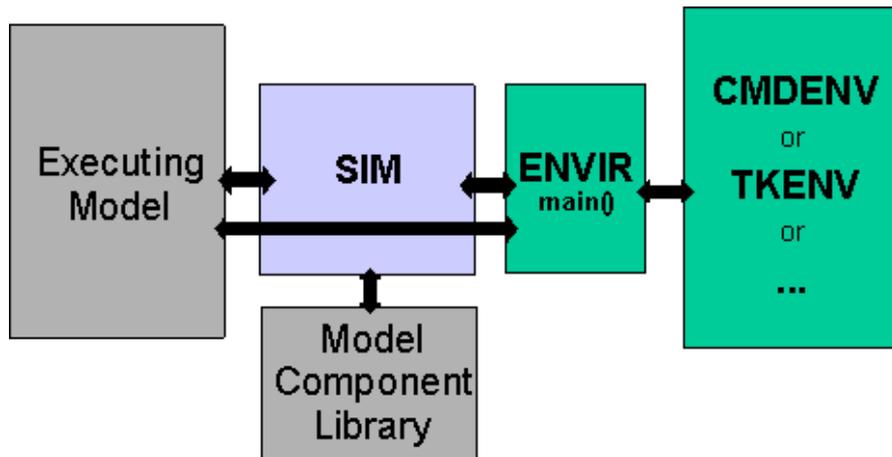


Figura D.3 Arquitectura de los programas de simulación en OMNET++, Fuente [3], OMNET++ Manual año 2005

- El Executing Model es el modelo que ha sido configurado para la simulación. Este contiene objetos que vienen a ser instancias de componentes dentro de la librería de modelos.

Las flechas de la figura D.3 muestran cómo interactúan los componentes entre ellos:

- Executing Model vs Sim. El kernel del simulador administra los eventos futuros e invoca a los módulos dentro del modelo de ejecución conforme los eventos van ocurriendo. Los módulos del modelo de ejecución son almacenados en el objeto principal (main) de Sim, clase cSimulation. A su vez, el modelo de ejecución hace el llamado a las funciones dentro del kernel de simulación y utiliza las clases pertenecientes a la librería Sim.
- Sim vs Model Component Library. El kernel de simulación inicializa los módulos simples y otros componentes cuando el modelo de simulación se configura al inicio de la ejecución de la simulación. Este también refiere a los

componentes de la librería cuando se usa creación dinámica de módulos. La maquinaria para registrar y buscar componentes dentro de componentes de modelo es implementada como parte de Sim.

- Executing Model vs Envir. El objeto `ev`, que es parte de Envir, constituye la fachada como interface de usuario hacia el modelo de ejecución. El modelo usa `ev` para escribir anotaciones.
- Sim vs Envir. Envir comanda todo lo que sucede dentro del programa de simulación. Envir contiene a la función principal `main()` en donde inicia la ejecución. Envir determina que modelos deben ser configurados para la simulación e instruye a Sim hacer eso. Envir contiene al loop principal de la simulación (que determina el evento siguiente y la secuencia ejecutar evento) e invoca al kernel de simulación por la funcionalidad necesaria (la programación de eventos y la ejecución de eventos son implementadas dentro de Sim). Envir captura y administra los errores, excepciones ocurrentes dentro del kernel de simulación o en la librería de clases durante la ejecución. Envir presenta una interface simple hacia el usuario (`ev`) que representa el ambiente (interface de usuario) hacia Sim. Durante la configuración del modelo de simulación, Envir proporciona a Sim los valores para los parámetros cada vez que Sim lo requiere. Por su parte, Sim escribe los vectores de salida vía Envir. Sim y sus clases usan Envir para imprimir información detallada (debug).
- Envir vs Tkenv y Cmdenv. Envir define `TOmnetApp` como clase de base para las interfaces de usuario, y define a `Tkenv` y `Cmdenv` como subclases de `TOmnetApp`. La función principal `main()` proporcionada como parte de Envir, determina la clase para interface de usuario apropiada, crea una instancia y la ejecuta. Las llamadas a Sim o al modelo dentro del objeto `ev` son simplemente derivadas hacia la instancia de `TOmnetApp`. Envir presenta una estructura (framework) y funcionalidad básica a `Tkenv` y `Cmdenv` por medio de métodos a `TOmnetApp` y algunas otras clases.

ANEXO E

Software de la Tesis

En este anexo se proporcionan los detalles del software utilizado para obtener los resultados de esta tesis: “Un protocolo MAC (control de acceso al medio) eficiente para redes inalámbricas”

Se adjunta un CD que contiene los programas fuente del protocolo MAC eficiente (ECRDA), y los archivos de configuración usados para ejecutar las simulaciones. A continuación se explica el esquema del CD y como utilizarlo.

E.1 Contenido del CD

El CD consta de dos carpetas:

- Carpeta ecrda
- Carpeta omnetpp.ini

Carpeta ecrda: contiene los archivos fuente del protocolo MAC, programados en C++. Una breve descripción de los archivos que contiene esta carpeta se presenta a continuación:

- Archivo readme.txt, brinda una descripción del programa fuente.
- Archivo phy.cc, define el comportamiento de la capa física para los módulos de terminal móvil y estación base.
- Archivo mac.cc, define el comportamiento de la capa mac para los módulos de terminal móvil y estación base.
- Archivo traffic.cc, define el comportamiento de la capa de tráfico para los módulos de terminal móvil y estación base
- Archivo network.ned, define la topología de la red ad-hoc simulada.
- Archivo messages.msg, define las estructuras de la trama CRDA.
- Archivo mobile.txt, define los parámetros de los terminales móviles.

Carpeta omnetpp.ini: contiene los archivos de las configuraciones correspondientes a cada simulación. El archivo de configuración se llama

omnetpp.ini. A los archivos contenidos en esta carpeta se les identifica como 1xxx-omnetpp.ini, donde 1xxx indica el ID de la simulación.

E.2 Simulaciones de la Tesis

Tabla E.1 Simulaciones de la Tesis

ID de la simulación	Descripción	Protocolo usado
1001	Simulación de 01 móvil de voz y 01 móvil de datos durante 27,000 segundos	CRDA
1002	Simulación de 80 móviles de voz y 40 móviles de datos durante 1,000 segundos	CRDA
1003	Simulación de 80 móviles de voz y 40 móviles de datos durante 500 segundos	CRDA
1004	Simulación de 80 móviles de voz y 40 móviles de datos durante 200 segundos	CRDA
1005	Simulación de 80 móviles de voz y 40 móviles de datos durante 100 segundos	CRDA
1006	Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de 0.0100 exponencial	CRDA
1007	Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de 0.0095 exponencial	CRDA
1008	Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de 0.0090 exponencial	CRDA
1009	Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de 0.0087 exponencial	CRDA
1010	Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de 0.0085 exponencial	CRDA
1011	Simulación de 100 móviles de voz y 40 móviles de datos con tiempo entre arribo de 0.0080 exponencial	CRDA
1012	Simulación de 100 móviles de voz y 20 de datos	CRDA

1013	Simulación de 100 móviles de voz y 30 móviles de datos	CRDA
1014	Simulación de 100 móviles de voz y 35 móviles de datos	CRDA
1015	Simulación de 100 móviles de voz y 40 móviles de datos	CRDA
1016	Simulación de 100 móviles de voz y 42 móviles de datos	CRDA
1017	Simulación de 100 móviles de voz y 44 móviles de datos	CRDA
1018	Simulación de 100 móviles de voz, 40 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 25 de clase de fondo.	CRDA
1019	Simulación de 100 móviles de voz, 20 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 5 de clase de fondo.	CRDA
1020	Simulación de 100 móviles de voz, 40 móviles de datos: 10 de clase conversacional, 10 de clase interactiva, 10 de clase unidireccional y 10 de clase de fondo.	ECRDA – protocolo MAC eficiente
1021	Simulación de 100 móviles de voz, 40 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 25 de clase de fondo.	ECRDA – protocolo MAC eficiente
1022	Simulación de 100 móviles de voz, 20 móviles de datos: 5 de clase conversacional, 5 de clase interactiva, 5 de clase unidireccional y 5 de clase de fondo.	ECRDA – protocolo MAC eficiente

E.3 Compilación del programa

Para compilar el programa ejecute las siguientes instrucciones:

1. Copie el contenido de la carpeta ECRDA dentro de su disco duro

Compilación como CRDA

Si desea ejecutar las simulaciones con CRDA entonces ingrese al archivo mac.cc y cambie la definición de QOS por el valor 0 en:

```
#define QOS 0
```

En caso contrario no cambie nada.

2. Abra un shell de Linux y vaya a la ubicación de la carpeta y ejecute:

- Para ejecución en modo consola
\$...\ecrda\opp_makemake -u Cmdenv
- Para ejecución en modo gráfico
\$...\ecrda\opp_makemake

3. Compile el programa con:

```
$...\ecrda\make
```

Se creará un ejecutable con el nombre de la carpeta que se llamará ecrda.

E.4 Ejecución del programa

Todas las simulaciones fueron ejecutadas usando el modo consola.

Para ejecutar las simulaciones debe copiar el archivo 111x-omnetpp.ini, que está ubicado en la carpeta ecrda. Luego deberá renombrar dicho archivo con omnetpp.ini.

Para ejecutar la simulación escriba:

```
$...\ecrda\./ecrda
```

E.5 Colección de resultados

Cuando termine la simulación se crearán dos archivos: omnetpp.vec y omnetpp.sca. Puede usar ambos archivos para el análisis de los resultados.

E.6 CD

<se adjunta el CD>