

ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA



Ing. NEIL ANDRÉS JIMÉNEZ PINZÓN

Director:

PhD. Jose Giovanni López Perafán

Universidad del Cauca

Facultad De Ingeniería Electrónica y Telecomunicaciones

Maestría en Electrónica y Telecomunicaciones

Departamento de Telecomunicaciones

Grupo I+D Nuevas Tecnologías en Telecomunicaciones – GNTT

Popayán, Junio de 2019

NEIL ANDRÉS JIMÉNEZ PINZÓN

**ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA
Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED
OBS DISTRIBUIDA**

**Trabajo de Grado presentado a la Facultad de
Ingeniería Electrónica y Telecomunicaciones de
la Universidad del Cauca para la obtención del
Título de**

**Magister en:
Electrónica y Telecomunicaciones**

**Director:
PhD. Jose Giovanni López Perafán**

Popayán

2019



CONTENIDO

INTRODUCCIÓN

CAPÍTULO 1. ASPECTOS BÁSICOS DE LAS REDES ÓPTICAS, REDES DE CONMUTACIÓN ÓPTICA Y TÉCNICAS DE CONMUTACIÓN, ARQUITECTURA Y ESQUEMA FUNCIONAL DE UNA RED OBS, PROCESO DE ENSAMBLE DE RÁFAGAS, FUNDAMENTOS Y CLASIFICACIÓN DE LOS ALGORITMOS DE ENSAMBLE..... 3

1. 1 Aspectos Básicos de las Redes Ópticas..... 3

1. 2 Redes de Conmutación Óptica –Técnicas 7

1.2 1 Conmutación Óptica de Circuitos OCS (Optical Circuit Switching) 8

1.2 2 Conmutación Óptica de Paquetes OPS (Optical Packet Switching)..... 9

1.2 3 Conmutación Óptica de Ráfagas OBS (Optical Burst Switching) 10

1. 3 Redes de Conmutación Óptica de Ráfagas 11

1.3. 1 Arquitectura y Esquema Funcional de una Red OBS 12

1.3. 2 Proceso de Ensamble de Ráfagas -Módulos de Ensamblaje 23

1. 4 Algoritmos de Ensamble de Ráfagas 26

1.4. 1 Algoritmo de ensamble basado en el tiempo –T..... 27

1.4. 2 Algoritmo de ensamble basado en el umbral de longitud –uL..... 27

1.4. 3 Algoritmo de ensamble mixto 27

1.4. 4 Algoritmos de ensamble de ráfagas híbridos dinámicos basados en calidad de servicio y por servicios diferenciados..... 28

1.4. 5 Algoritmos de ensamble de ráfagas híbridos dinámicos por predicción 28

1. 5 Clasificación de los Algoritmos de Ensamble 29

CAPÍTULO 2. CLASES DE PROBLEMAS SEGÚN LA TEORÍA DE LA COMPLEJIDAD, FUNDAMENTOS Y CLASES DE METAHEURÍSTICAS, INTRODUCCIÓN A LA METAHEURÍSTICA POR OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS Y ELEMENTOS, OPERACIONES Y PRINCIPIO BÁSICO DE LA TÉCNICA LÓGICA DIFUSA..... 31

2. 1 Clases de Problemas –Teoría de la Complejidad..... 31

2. 2 Metaheurísticas -Fundamentos 32

2.2. 1 Metaheurísticas basadas en trayectoria 35



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

2.2. 2 Metaheurísticas basadas en población	36
2. 3 Técnica Lógica Difusa	39
2.3.1 Número Difuso, Conjunto Difuso e Intervalo Difuso	40
2.3.2 Función de Pertenencia	40
2.3.3 Variables Lingüísticas, Variable Difusa y Estado Difuso	43
2.3.4 Operaciones entre Conjuntos Difusos	43
2.3.5 Principio básico de la técnica lógica difusa	45
CAPÍTULO 3. METODOLOGÍA DE DESARROLLO Y DE SIMULACIÓN, SELECCIÓN DE LA HERRAMIENTA DE SIMULACIÓN, TOPOLOGIA A IMPLEMENTAR EN LA RED OBS DE PRUEBA Y SUS MÓDULOS, DIAGRAMA DE FLUJO DEL ALGORITMO DE ENSAMBLE MIXTO Y DEL DMBAA-FLPSO Y PSEUDOCÓDIGO DEL DMBAA-FLPSO	
3. 1 Metodología	49
3.1.1 Metodología de Desarrollo	49
3.1.2 Metodología de Simulación	50
3. 2 Selección de la Herramienta de Simulación.....	52
3.2. 1 NS-2 (Network Simulator 2)	52
3.2. 2 NS-3 (Network Simulator 3)	52
3.2. 3 OPNET	53
3.2. 4 NCTUNS (National Chiao Tung University Network Simulator).....	53
3.2. 5 OMNeT++	53
3.2. 6 Comparación entre las diferentes herramientas de simulación	54
3. 3 Módulos DE una Red OBS	56
3.3. 1 Nodo Fuente / Destino.....	56
3.3. 2 Nodo Borde (Edge Node)	56
3.3. 3 Nodo Central (Core Node).....	58
3.3. 4 Red OBS basada en la topología NSFNeT	60
3. 4 DIAGRAMA DE FLUJO DEL Algoritmo de Ensamble mixto y DEL DMBAA-FLPSO y pseudocódigo DEL DMBAA-FLPSO	61
CAPÍTULO 4. ESCENARIOS Y CASOS DE SIMULACIÓN, PRUEBAS PARA LA EVAVUACIÓN Y ANALISIS DEL DESEMPEÑO DEL ALGORITMO DMBAA-FLPSO CON EL ALGORITMO DE ENSAMBLE MIXTO....	
4. 1 Introducción	68



4. 2 Casos de Simulación	68
4.2. 1 Caso de Estudio 1: Red OBS–NSFNeT Distribuida con el Algoritmo de Ensamble de Ráfagas Mixto (clásico).....	68
4.2. 2 Caso de Estudio 2: Red OBS–NSFNeT Distribuida con el Algoritmo de Ensamble de Ráfagas Mixto Híbrido Adaptativo (propuesto)	69
4.2. 3 Sub Casos de Simulación	69
4. 3 Pruebas de Simulación.....	71
4. 4 Parámetros A ConfiguraR EN los Escenarios de Simulación	83
4. 5 Cantidad total de Simulaciones y su ejecución	85
4. 6 ANÁLISIS DE RESULTADOS	87
4.6 1 Pruebas de Simulación para Carga Muy Baja, sin/con el Algoritmo DMBAAs-FLPSO, con Semilla 108, Semilla 9, para $uL=(2, 3, 4)$ y $T=(2.75, 2.5, 2.25)$. Determinando la PB_ráfagas y el e2e-D_ráfagas.....	87
4.6 2 Pruebas de Simulación para Carga Baja, sin/con el Algoritmo DMBAAs-FLPSO, con Semilla 40187 /Semilla 7385 y para $uL=(2, 3, 4)$ y $T=(2.75, 2.5, 2.25)$. Determinando PB_ráfagas y e2e-D_ráfagas.....	94
4.6 3 Pruebas de Simulación para Carga Media, sin/con el Algoritmo DMBAAs-FLPSO, con Semilla 33 /Semilla 297834 y para $uL=(3, 4, 5)$ y $T=(2.75, 2.5, 2.25)$. Determinando PB_ráfagas y e2e-D_ráfagas	101
4.6 4 Pruebas de Simulación para Carga Alta, sin/con el Algoritmo DMBAAs-FLPSO, con Semilla 532570 /Semilla 150 y para $uL=(4, 5, 6)$ y $T=(2.75, 2.5, 2.25)$. Determinando PB_ráfagas y e2e-D_ráfagas.....	108
4.6 5 Pruebas de Simulación para Carga Muy Alta, sin/con el Algoritmo DMBAAs-FLPSO, con Semilla 70 /Semilla 20 y para $uL=(4, 5, 6)$ y $T=(2.75, 2.5, 2.25)$. Determinando PB_ráfagas y e2e-D_ráfagas.....	115
CAPÍTULO 5 CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS	132
5. 1 CONCLUSIONES	132
5.1. 1 Respecto a los resultados de simulación	132
5.1. 2 Respecto al desarrollo del Trabajo de Grado de Maestría.....	133
5.1. 3 Respecto a la herramienta de simulación.....	134
5. 2 RECOMENDACIONES	134
5. 3 TRABAJOS FUTUROS	135
REFERENCIAS	136



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

ANEXO A. ALGORITMOS

ANEXO B. RESULTADOS DE SIMULACIÓN PARA 5.0 Gbps



LISTA DE FIGURAS

Figura 1. 1. Incremento en la suscripción a Internet fijo y móvil 1T-2017 al 1T-2018.....	3
Figura 1. 2. Incremento en la suscripción a Internet fijo y móvil 4T-2015 al 4T-2016.....	4
Figura 1. 3. Atenuación vs Longitud de Onda por frecuencia. Pérdidas Ópticas.....	4
Figura 1. 4. Atenuación vs Longitud de Onda según su Ventana de Transmisión y sus Bandas	4
Figura 1. 5 Esquema general de una red óptica WDM	5
Figura 1. 6 Espaciamiento entre canales y Bandas del espectro óptico en CWDM y DWDM.....	6
Figura 1. 7 Conversión OXC	7
Figura 1. 8 Red de Conmutación Óptica de Circuitos OCS	9
Figura 1. 9 Red de Conmutación Óptica de Paquetes OPS.....	9
Figura 1. 10 Esquema de transmisión del BCP y de la Ráfaga	11
Figura 1. 11 Diagrama de tiempo para el envío de la ráfaga y de su BCP.....	12
Figura 1. 12 Arquitectura general de una red de comunicación óptica.....	13
Figura 1. 13 Clasificación de los Nodos de Red según el Esquema funcional de una red OBS.....	14
Figura 1. 14 Modelo de Transmisión según el Esquema funcional en una red OBS.....	14
Figura 1. 15 Ensamble de Ráfagas en un Nodo Borde de Entrada	15
Figura 1. 16 Arquitectura de un Nodo Borde de Entrada OBS	15
Figura 1. 17 Desensamble de Ráfaga en un Nodo Borde de Salida	16
Figura 1. 18 Tipos de Señalización OBS (a) centralizada con two-way y (b) distribuida con one-way	17
Figura 1. 19 Protocolo de Señalización JET	18
Figura 1. 20 Arquitectura de un Nodo Central OBS	21
Figura 1. 21 Conmutación de Ráfagas en un Nodo Borde de Entrada.....	22
Figura 1. 22 Conmutación de Ráfagas en un Nodo Core.....	23
Figura 1. 233 Módulos del esquema Ensamble de Ráfagas	23
Figura 1. 244 Módulos del proceso de Ensamble en un Nodo Borde de Entrada OBS.....	24
Figura 1. 25 Esquema de Ensamble de Ráfagas –Parámetros de Ensamble	26



Figura 2. 1 Clases de Problemas según la Teoría de la Complejidad	32
Figura 2. 2 Clasificación de los Métodos de Optimización	33
Figura 2. 3 Clasificación de las metaheurísticas	34
Figura 2. 4 Número difuso e Intervalo difuso.....	40
Figura 2. 5 Función de Pertenencia Triangular.....	41
Figura 2. 6 Función de Pertenencia Trapezoidal.....	41
Figura 2. 7 Función de Pertenencia Trapezoidal $a = b = -\infty$	42
Figura 2. 8 Función de Pertenencia Trapezoidal $c = d = +\infty$	42
Figura 2. 9 Función de Pertenencia Sigmoidal.....	42
Figura 2. 10 Conceptos lingüísticos en cada una de las variables difusas y la función de pertenencia trapezoidal.....	43
Figura 2. 11 Unión entre A y B y Complemento de A	44
Figura 2. 12 Intersección entre A y B.	44
Figura 2. 13 Operación unión $(A \cap B)$, para determinar la función de Pertenencia del estado difuso A2	45
Figura 2. 14 Esquema funcional de la técnica Lógica Difusa	45
Figura 2. 15 Cantidad de paquetes entrantes a un nodo borde OBS vs el grado de pertenencia a un conjunto bajo la lógica convencional.....	47
Figura 2. 16 Cantidad de paquetes entrantes a un nodo borde OBS vs el grado de pertenencia a un conjunto bajo la lógica difusa.....	47
Figura 3. 1 Metodología aplicada para el desarrollo del trabajo de grado de Maestría.....	49
Figura 3. 2 Metodología aplicada para la Simulación de los Escenarios (casos) y SubCasos implementando el Algoritmo de Ensamble Clásico y Propuesto.....	51
Figura 3. 3 Modulo Fuente / Destino en OMNeT++. En un nodo borde OBS.....	56
Figura 3. 4 Módulo del Nodo Borde de Entrada y de Salida	57
Figura 3. 5 Submódulos del Módulo de Ensamble en el Nodo Borde de Entrada.....	57
Figura 3. 6 Conexión entre el Nodo Borde y el Nodo Central en una red OBS.....	59
Figura 3. 7 Módulos en un Nodo Central	59
Figura 3. 8 Módulos de Unidad de Control en un Nodo Central	60
Figura 3. 9 Topología de la NSFNET	60



Figura 3. 10 Red OBS basada en la topología NSFNet –implementada en OMNeT	61
Figura 3. 11 Diagrama de Flujo General del Algoritmo de Ensamble Mixto –Clásico.....	62
Figura 3. 12 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Nodo de Borde de Entrada OBS y Módulo 1 Clasificación	63
Figura 3. 13 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Módulo 2 Ensamblaje	63
Figura 3. 14 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Módulo 3 Planificación	64
Figura 3. 15 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Nodo Borde de Salida OBS.....	64
Figura 4. 1 Conjuntos Difusos uL utilizando la técnica Lógica Difusa con Función Trapezoidal para Carga Muy Baja, Carga Baja, Carga Media, Carga Alta y Carga Muy Alta	72
Figura 4. 2 Valores uL utilizando la técnica Lógica Difusa con Función Triangular para Carga Muy Baja y Carga Baja.....	73
Figura 4. 3 Valores uL utilizando la técnica Lógica Difusa con Función Triangular para Carga Muy Media	73
Figura 4. 4 Valores uL utilizando la técnica Lógica Difusa con Función Triangular para Carga Muy Alta y Carga Alta.....	73
Figura 4. 5 Pruebas de Simulación para Carga Muy Alta con Semilla 20 y 70 con uL=(4, 5, 6) y T=(2.25, 2.5, 2.75)	81
Figura 4. 6 Pruebas de Simulación para Carga Alta con Semilla 150 y 532570 con uL=(4, 5, 6) con T=(2.25, 2.5, 2.75)	81
Figura 4. 7 Pruebas de Simulación para Carga Media con Semilla 297834 y 33 con uL=(3, 4, 5) con T=(2.25, 2.5, 2.75)	82
Figura 4. 8 Pruebas de Simulación para Carga Baja con Semilla 7385 y 40187 con uL=(3, 4, 5) con T=(2.25, 2.5, 2.75)	82
Figura 4. 9 Pruebas de Simulación para Carga Muy Baja con Semilla 9 y 108 con uL=(2, 3, 4) con T=(2.25, 2.5, 2.75)	83
Figura 4. 10 PB por nodo para Carga Muy Baja con 0.1E, con uL=2/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.	87
Figura 4. 11 PB por nodo para Carga Muy Baja con 0.1E, con uL=3/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.	88
Figura 4. 12 PB por nodo para Carga Muy Baja con 0.1E, con uL=4/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.	88



Figura 4. 13 PB nodo 6 para Carga Muy Baja con 0.1E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	89
Figura 4. 14 PB nodo 7 para Carga Muy Baja con 0.1E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	89
Figura 4. 15 PB promedio por red para Carga Muy Baja con 0.1E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.....	90
Figura 4. 16 PB por nodo para Carga Muy Baja con 0.2E, con $uL=2/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	91
Figura 4. 17 PB por nodo para Carga Muy Baja con 0.2E, con $uL=3/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	91
Figura 4. 18 PB por nodo para Carga Muy Baja con 0.4E, con $uL=2/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	92
Figura 4. 19 PB nodo 6 para Carga Muy Baja con 0.2E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	92
Figura 4. 20 PB nodo 7 para Carga Muy Baja con 0.2E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	93
Figura 4. 21 PB promedio por red para Carga Muy Baja con 0.2E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.....	93
Figura 4. 22 PB por nodo para Carga Baja con 0.3E, con $uL=2/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	94
Figura 4. 23 PB por nodo para Carga Baja con 0.3E, con $uL=3/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	95
Figura 4. 24 PB por nodo para Carga Baja con 0.3E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	95
Figura 4. 25 PB nodo 6 para Carga Baja con 0.3E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	96
Figura 4. 26 PB nodo 7 para Carga Baja con 0.3E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	96
Figura 4. 27 PB promedio por red para Carga Baja con 0.3E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	97
Figura 4. 28 PB por nodo para Carga Baja con 0.4E, con $uL=2/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	98
Figura 4. 29 PB por nodo para Carga Baja con 0.4E, con $uL=3/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	98



Figura 4. 30 PB por nodo para Carga Baja con 0.4E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	99
Figura 4. 31 PB nodo 6 para Carga Baja con 0.4E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	99
Figura 4. 32 PB nodo 7 para Carga Baja con 0.4E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	100
Figura 4. 33 PB promedio por red para Carga Baja con 0.4E, con $uL=(2, 3, 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	100
Figura 4. 34 PB por nodo para Carga Media con 0.5E, con $uL=3/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	101
Figura 4. 35 PB por nodo para Carga Media con 0.5E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	102
Figura 4. 36 PB por nodo para Carga Media con 0.5E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	102
Figura 4. 37 PB nodo 6 para Carga Media con 0.5E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	103
Figura 4. 38 PB nodo 7 para Carga Media con 0.5E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	103
Figura 4. 39 PB promedio por red para Carga Media con 0.5E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	104
Figura 4. 40 PB por nodo para Carga Media con 0.6E, con $uL=3/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.	105
Figura 4. 41 PB por nodo para Carga Media con 0.6E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.	105
Figura 4. 42 PB por nodo para Carga Media con 0.6E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.	106
Figura 4. 43 PB nodo 6 para Carga Media con 0.6E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.	106
Figura 4. 44 PB nodo 7 para Carga Media con 0.6E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.	107
Figura 4. 45 PB promedio por red para Carga Media con 0.6E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.	107
Figura 4. 46 PB por nodo para Carga Alta con 0.7E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	108



Figura 4. 47 PB por nodo para Carga Alta con 0.7E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	109
Figura 4. 48 PB por nodo para Carga Alta con 0.7E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	109
Figura 4. 49 PB nodo 6 para Carga Alta con 0.7E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	110
Figura 4. 50 PB nodo 7 para Carga Alta con 0.7E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	110
Figura 4. 51 PB promedio por red para Carga Alta con 0.7E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	111
Figura 4. 52 PB por nodo para Carga Alta con 0.8E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	112
Figura 4. 53 PB por nodo para Carga Alta con 0.8E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	112
Figura 4. 54 PB por nodo para Carga Alta con 0.8E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	113
Figura 4. 55 PB nodo 6 para Carga Alta con 0.8E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	113
Figura 4. 56 PB nodo 7 para Carga Alta con 0.8E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	114
Figura 4. 57 PB promedio por red para Carga Alta con 0.8E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.	114
Figura 4. 58 PB por nodo para Carga Muy Alta con 0.9E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	115
Figura 4. 59 PB por nodo para Carga Muy Alta con 0.9E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	116
Figura 4. 60 PB por nodo para Carga Muy Alta con 0.9E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	116
Figura 4. 61 PB nodo 6 para Carga Muy Alta con 0.9E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	117
Figura 4. 62 PB nodo 7 para Carga Muy Alta con 0.9E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	117
Figura 4. 63 PB promedio por red para Carga Muy Alta con 0.9E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	118



Figura 4. 64 PB por nodo para Carga Muy Alta con 1E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	119
Figura 4. 65 PB por nodo para Carga Muy Alta con 1E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico	119
Figura 4. 66 PB por nodo para Carga Muy Alta con 1E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	120
Figura 4. 67 PB nodo 6 para Carga Muy Alta con 1E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	120
Figura 4. 68 PB nodo 7 para Carga Muy Alta con 1E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.	121
Figura 4. 69 PB promedio por red para Carga Muy Alta con 1E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.....	121
Figura 4. 70 Probabilidad de Bloqueo vs Tráfico para 2.5Gbps Con/Sin Algoritmo DMBAA-FLPSO.	125
Figura 4. 71 Porcentaje de Ráfagas Perdidas vs Tráfico para 2.5Gbp Con/Sin Algoritmo DMBAA-FLPSO.	125
Figura 4. 72 Retardo extremo a extremo ($e2e-D_{ráfagas}$) vs Tráfico para 2.5Gbps Con/Sin Algoritmo DMBAA-FLPSO.....	126
Figura 4. 73 Nivel de Desempeño del Algoritmo DMBAA-FLPSO en cuanto a la $PB_{ráfagas}$ con un enlace de 2.5Gbps.....	130



LISTA DE TABLAS

Tabla 1. 1 Parámetros típicos en redes CWDM y DWDM	6
Tabla 1. 2 Clases de Granularidad en la Conmutación en redes ópticas	8
Tabla 1. 3 Características principales de la Conmutación Óptica OCS, OPS y OBS	10
Tabla 1. 4 Clasificación de los Módulos del proceso de Ensamble de Ráfagas.	25
Tabla 1. 5 Lista de acrónimos de la tabla 1.6.....	29
Tabla 1. 6 Clasificación de algunos de los Algoritmos, Técnicas y Mecanismos basados en el Ensamble de Ráfagas	30
Tabla 3. 1 Requerimientos analizados sobre las herramientas de simulación.....	55
Tabla 4. 1 Escenario y Casos de Estudio de Simulación	70
Tabla 4. 2 Pruebas de Simulación para Carga Muy Alta con Semilla 20, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.....	74
Tabla 4. 3 Pruebas de Simulación para Carga Muy Alta con Semilla 70, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.....	74
Tabla 4. 4 Pruebas de Simulación para Carga Alta con Semilla 150, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.....	75
Tabla 4. 5 Pruebas de Simulación para Carga Alta con Semilla 532570, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.....	76
Tabla 4. 6 Pruebas de Simulación para Carga Media con Semilla 297834, para la combinación del Sub Caso 3, Sub Caso 4 y Sub Caso 5 con los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3, y Sub Casos 5.1, 5.2 y 5.3 respectivamente.....	76
Tabla 4. 7 Pruebas de Simulación para Carga Media con Semilla 33, para la combinación del Sub Caso 3, Sub Caso 4 y Sub Caso 5 con los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3, y Sub Casos 5.1, 5.2 y 5.3 respectivamente.....	77
Tabla 4. 8 Pruebas de Simulación para Carga Baja con Semilla 7385, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.....	78



Tabla 4. 9 Pruebas de Simulación para Carga Baja con Semilla 40187, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.....	78
Tabla 4. 10 Pruebas de Simulación para Carga Muy Baja con Semilla 9, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.....	79
Tabla 4. 11 Pruebas de Simulación para Carga Muy Baja con Semilla 108, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.....	80
Tabla 4. 12 Parámetros a configurar en la red OBS NSFNeT Distribuida	83
Tabla 4. 13 Especificaciones técnicas de los equipos utilizados.....	85
Tabla 4. 14 Carga de Tráfico según el Tiempo entre Paquetes (T_p) y el Tiempo de Simulación (T_s)	86
Tabla 4. 15 PB_ráfagas para Carga Muy Baja 0.1E con y sin el algoritmo DMBAА-FLPSO.....	90
Tabla 4. 16 PB_ráfagas para Carga Muy Baja 0.2E con y sin el algoritmo DMBAА-FLPSO.....	94
Tabla 4. 17 PB_ráfagas para Carga Baja 0.3E con y sin el algoritmo DMBAА-FLPSO.....	97
Tabla 4. 18 PB_ráfagas para Carga Baja 0.4E con y sin el algoritmo DMBAА-FLPSO.....	101
Tabla 4. 19 PB_ráfagas para Carga Baja 0.5E con y sin el algoritmo DMBAА-FLPSO.....	104
Tabla 4. 20 PB_ráfagas para Carga Baja 0.6E con y sin el algoritmo DMBAА-FLPSO.....	108
Tabla 4. 21 PB_ráfagas para Carga Baja 0.7E con y sin el algoritmo DMBAА-FLPSO.....	111
Tabla 4. 22 PB_ráfagas para Carga Baja 0.8E con y sin el algoritmo DMBAА-FLPSO.....	115
Tabla 4. 23 PB_ráfagas para Carga Baja 0.9E con y sin el algoritmo DMBAА-FLPSO.....	118
Tabla 4. 24 PB_ráfagas para Carga Baja 1E con y sin el algoritmo DMBAА-FLPSO.....	122
Tabla 4. 25 PB_ráfagas y e2e-D_ráfagas para Carga Muy Alta.....	122
Tabla 4. 26 PB_ráfagas y e2e-D_ráfagas para Carga Alta.....	123
Tabla 4. 27 PB_ráfagas y e2e-D_ráfagas para Carga Media.....	123
Tabla 4. 28 PB_ráfagas y e2e-D_ráfagas para Carga Baja.....	124
Tabla 4. 29 PB_ráfagas y e2e-D_ráfagas para Carga Muy Baja.....	124
Tabla 4. 30 Síntesis de los datos-resultados Finales de la PB_ráfagas y el e2e-D para 2.5Gbps al implementar el Algoritmo DMBAА-FLPSO y el Algoritmo de Ensamble Mixto Clásico	127
Tabla 4. 31 Síntesis de los datos-resultados Finales en porcentaje de la PB_ráfagas y el e2e-D para 2.5Gbps al implementar el Algoritmo DMBAА-FLPSO y el Algoritmo de Ensamble Mixto Clásico	128



Tabla 4. 32 Diferencia en porcentaje de la PB_ráfagas entre el algoritmo DMBAA-FLPSO y el algoritmo de ensamble Clásico..... 129



LISTA DE ACRÓNIMOS

ACO	<i>Ant Colony Optimization</i> , Optimización basada en Colonias de Hormigas.
ATM	<i>Asynchronous Transfer Mode</i> , Modo de Transferencia Asíncrono.
BCP	<i>Burst Control Packet</i> , Paquete de Control de la Ráfaga.
BD	<i>Buffer Destination</i> , Buffer's Destino.
CU	<i>Control Unit</i> , Unidad de Control.
CWDM	<i>Coarse Wavelength División Multiplexing</i> , Multiplexación por División Aproximada de Longitud de Onda.
DiffServ	<i>Differentiated Services</i> , Servicios Diferenciados.
DW	<i>Distance Weight</i> , Menor Distancia.
DWDM	<i>Dense Wavelength División Multiplexing</i> , Multiplexación por División de Longitud de Onda Densa.
e2e-D	<i>End-to-End Delay</i> , Retardo extremo a extremo.
EA	<i>Evolutionary Algorithms</i> , Algoritmos Evolutivos.
EO	<i>Electronic-Optical</i> , Conversión Eléctrico-Óptico.
ERaf	Ensamble de Ráfagas.
FDL	<i>Fiber Delay Line</i> , Línea de Retardo de Fibra.
FF	<i>First Fit</i> , Primer Ajuste.
FL	<i>Fuzzy Logic</i> , Lógica Difusa.
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i> , Procedimiento de Búsqueda Miope Aleatorio y Adaptativo.
HW	<i>Hop Weight</i> , Peso por Salto.
ILS	<i>Iterated Local Search</i> , Búsqueda Local Iterada.
IP	<i>Internet Protocol</i> , Protocolo de Internet.
JET	<i>Just Enough Time</i> , Suficiente Tiempo.
JIT	<i>Just In Time</i> , Justo a Tiempo.
LAUC	<i>Lastet Available Unscheduled Channel</i> , Último Canal no Planificado Disponible.
LU	<i>Least Used</i> , Menos Usado.
MU	<i>Most Used</i> , Más Usado.
NSFNeT	<i>National Science Foundation Network</i> , Red de la Fundación Nacional para la Ciencia.



OBS	<i>Optical Burst Switching</i> , Conmutación Óptica de Ráfagas.
OCS	<i>Optical Circuit Switching</i> , Conmutación Óptica de Circuitos.
OE	<i>Optical-Electronic</i> , Conversión Óptico-Eléctrico.
OEO	<i>Optical Electronic Optical</i> , Óptica-Eléctrico-Óptica.
OPS	<i>Optical Packets Switching</i> , Conmutación Óptica de Paquetes.
OXC	<i>Optical Cross-Connect</i> , Trans-conector Óptico.
PB	Probabilidad de Bloqueo.
PER	<i>Parameter of Burst Assembly</i> , Parámetros de Ensamble de Ráfagas.
PSO	<i>Particle Swarm Optimization</i> , Optimización por Enjambre de Partículas.
QoS	<i>Quality of Service</i> , Calidad de Servicio.
RWA	<i>Routing and Wavelength Assignment</i> , Encaminamiento y Asignación de Longitud de Onda.
SA	<i>Simulated Annealing</i> , Enfriamiento Simulado.
SDM	<i>Space Division Multiplexing</i> , Multiplexación por División de Espacio.
SP	<i>Shortest Path</i> , Camino más Corto.
SONET	<i>Synchronous Optical NETWORK</i> , Redes Ópticas Sincrónica.
SS	<i>Scatter Search</i> , Búsqueda Dispersa.
TAG	<i>Tell And Go</i> , Ve y Dile.
TCP	<i>Transmission Control Protocol</i> , Protocolo de Control de Transmisión.
TDM	<i>Time Division Multiplexing</i> , Multiplexación por División en el Tiempo.
TIC	Tecnologías de la Información y la Comunicación.
TS	<i>Tabu Search</i> , Búsqueda Tabú.
VNS	<i>Variable Neighborhood Search</i> , Búsqueda con vecindario variable.
WC	<i>Wavelength Converter</i> , Conversores de Longitud de Onda.
WDM	<i>Wavelength Division Multiplexing</i> , Multiplexación por División de Longitud de Onda.



INTRODUCCIÓN

En la última década, las redes de comunicaciones han pasado de ambientes de baja velocidad, soportados en redes tradicionales basadas en cobre, a ambientes de alta velocidad, soportados en redes basadas en fibra óptica. En la actualidad, las redes de comunicaciones han experimentado un crecimiento exponencial en el tráfico de datos, debido al aumento en el número de conexiones en la red y al desarrollo de nuevas aplicaciones. Estas últimas, con requerimientos más exigentes en cuanto a tiempo real y en ancho de banda.

Son las redes troncales, las encargadas de soportar y procesar esas conexiones de grandes dimensiones de tráfico de datos en sus enlaces y nodos. Por lo tanto, se hace necesario hablar de redes de banda ancha que pueda soportar dicho dimensionamiento y requerimientos; por ejemplo, en [1], el Boletín Trimestral de las TIC (*Tecnologías de la Información y la Comunicación*) reporta, en el primer trimestre de 2018 el número de conexiones a Internet de Banda Ancha en Colombia alcanzo un total de 30.383.982 millones, en contraste, en el primer trimestre de 2017 con un total de 28.393.254 millones y en el cuarto trimestre de 2017 con un total de 30.282.371 millones, donde la variación porcentual entre el 1T-2017 y el 1T-2018 es del 7.0% (incremento) y la variación porcentual entre el 4T-2017 y el 1T-2018 es del 0.3% (incremento).

Las redes troncales actuales están llegando al límite del uso de sus recursos en su diseño inicial de escalamiento, donde el dimensionamiento del tráfico presenta el problema de no poder satisfacer la gran demanda de ancho de banda por parte de la conexión de los usuarios. Para resolver este problema, la ingeniería en el área de las redes y las telecomunicaciones presenta a la tecnología de Conmutación Óptica por Ráfagas (OBS, *Optical Burst Switching*) como la más favorable actualmente y para la próxima generación de las redes totalmente ópticas.

Mejorar la eficiencia en los procesos involucrados en las redes OBS es un campo de investigación abierto, el cual se puede extraer datos e información a través del uso de técnicas, algoritmos y herramientas de simulación, basados en parámetros y criterios de la realidad, con el fin de dar respuesta a alguna situación o posible solución a un problema en las redes ópticas. Debido a que cada proceso cumple con una función de vital importancia para el desempeño de toda la red. Por esta razón, cada proceso es considerado como un problema de estudio a mejorar. Se destaca la importancia del proceso de ensamble en una red OBS, dado que tiene una gran incidencia en el funcionamiento sobre los otros procesos y por lo tanto sobre el desempeño de toda la red, ya que el ensamble de ráfagas es el encargado de realizar la administración del volumen del flujo de tráfico entrante a los nodos borde y a los nodos core (núcleo o centrales) de la red.

El presente trabajo, centra su estudio en el proceso de ensamble de ráfagas de datos de una red OBS, cuyo objetivo es proponer un esquema diferente de realizar



el ensamblaje, para determinar el valor más cercano al óptimo de los parámetros de ensamble y asignar el tamaño más adecuado a la ráfaga a transmitir (tamaño adaptativo), esto mediante el desarrollo de un nuevo Algoritmo de Ensamble Híbrido Mixto Adaptativo. Híbrido en cuanto a que integra la técnica Lógica Difusa y la Metaheurística Optimización por Enjambre de Partículas (PSO, *Particle Swarm Optimization*), llamado DMBAA-FLPSO (*Dynamic Mixed Burst Assembly Algorithms–Fuzzy Logic and PSO*).

Para la realización del trabajo de grado de Maestría y el abordaje de los temas necesarios para el mismo, el contenido se divide en cinco capítulos:

En el **capítulo 1**: se presentan los aspectos básicos de las redes ópticas, descripción de las redes de conmutación y comparación de las técnicas de conmutación óptica, introducción, arquitectura y esquema funcional de las redes de conmutación óptica de ráfagas OBS (*Optical Burst Switching*), fundamentos del proceso de ensamble de ráfagas de datos y sus módulos, descripción general y clasificación de algunos de los algoritmos de ensamble existentes en la literatura científica OBS. Este último, es el resultado del objetivo específico 1, en el presente trabajo de grado de Maestría.

En el **capítulo 2**: se presenta una descripción general de las diferentes clases de problemas según la teoría de complejidad, fundamentos y clases de metaheurísticas, introducción a la metaheurística por Optimización de Enjambre de Partículas (PSO) y los elementos como función de pertenencia, conjuntos difusos, variable lingüística, entre otros, operaciones y principio básico de la técnica lógica difusa.

En el **capítulo 3**: se define la metodología de desarrollo y de simulación, selección de la herramienta de simulación, topología a implementar en la red OBS de prueba y sus módulos, diseño del algoritmo de ensamble mixto (clásico) y del algoritmo de ensamble híbrido mixto adaptativo DMBAA-FLPSO (propuesto), presentando el diagrama de flujo de los dos algoritmos de ensamble y el pseudocódigo del DMBAA-FLPSO.

En el **capítulo 4**: Se define los escenarios y casos de simulación y se desarrolla el plan de pruebas de simulación usando la herramienta IDE OMNeT++, para la evaluación y análisis de desempeño del algoritmo de ensamble propuesto, DMBAA-FLPSO, sobre la red OBS-NSFNet Distribuida, en cuanto a la probabilidad de bloqueo o tasa de pérdidas (PB) y el retardo extremo a extremo (e2e-D, *End-to-End Delay*). Comparando los resultados de desempeño del algoritmo de ensamble DMBAA-FLPSO con los del algoritmo de ensamble mixto.

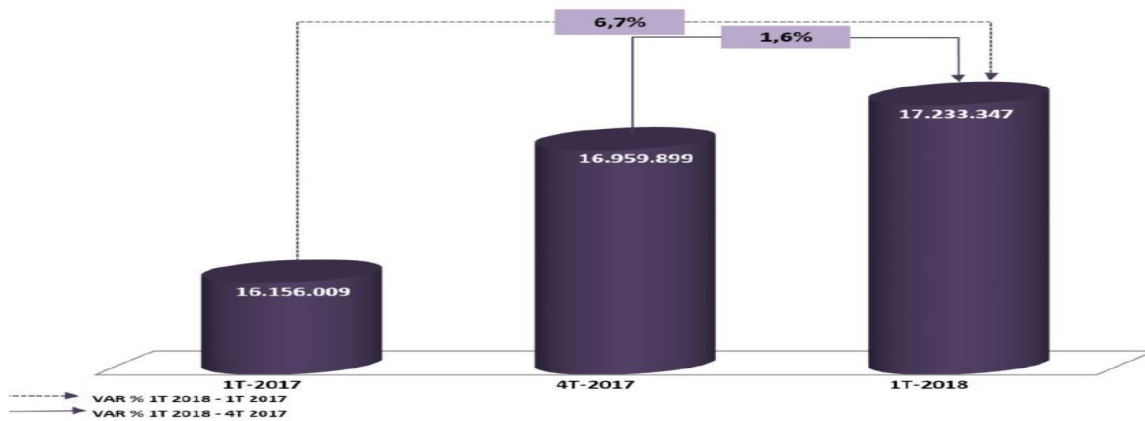
Finalmente, en el **capítulo 5**, se presentan las conclusiones del trabajo de grado de Maestría, recomendaciones y propuestas sobre posibles trabajos futuros relacionados en esta área, las redes de conmutación ópticas.



CAPÍTULO 1. ASPECTOS BÁSICOS DE LAS REDES ÓPTICAS, REDES DE CONMUTACIÓN ÓPTICA Y TÉCNICAS DE CONMUTACIÓN, ARQUITECTURA Y ESQUEMA FUNCIONAL DE UNA RED OBS, PROCESO DE ENSAMBLE DE RÁFAGAS, FUNDAMENTOS Y CLASIFICACIÓN DE LOS ALGORITMOS DE ENSAMBLE

1. 1 ASPECTOS BÁSICOS DE LAS REDES ÓPTICAS

Las redes troncales (o backbone) han evolucionado debido a los cambios presentados en la era actual, llamada “la era de la información”, basada en la transmisión de información y la comunicación. Dichos cambios, son impulsados por el crecimiento del volumen de la demanda de tráfico entrante a la red [1 - 2]; por ejemplo, la figura 1.1 y la figura 1.2, muestran el incremento y variación porcentual de suscripción a internet fijo y móvil en el 1T-2017 al 1T-2018, y en el 4T-2015 al 4T-2016 respectivamente. La principal característica en esta evolución es el medio de transmisión o enlace, donde han dado un salto de las redes troncales por líneas de cobre (redes tradicionales), a las redes troncales sobre fibra óptica (redes ópticas).



Fuente: Datos reportados por los Proveedores de Redes y Servicios a Colombia TIC.
Fecha de consulta: 6 de julio del 2018.

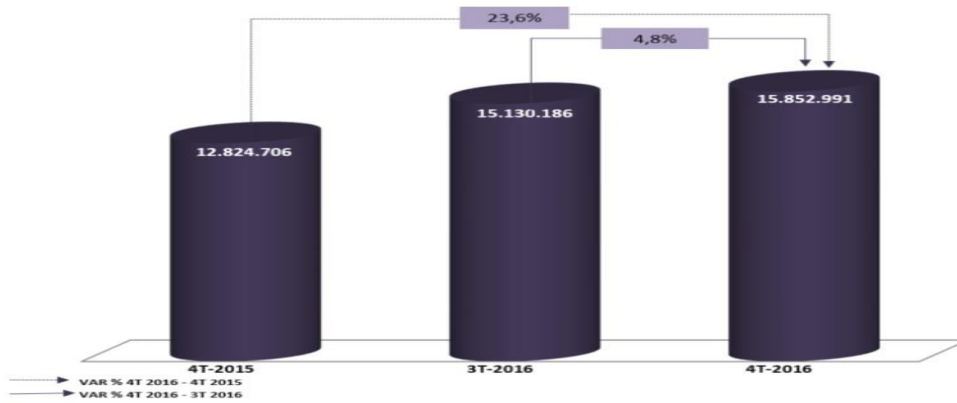
Figura 1. 1. Incremento en la suscripción a Internet fijo y móvil 1T-2017 al 1T-2018

Fuente: Tomado de [3] [4].

La fibra óptica presenta una serie de ventajas sobre las tecnologías convencionales, cabe mencionar su gran ancho de banda (recurso de red), definida como la capacidad de enviar mayor información en una misma unidad de tiempo. El ancho de banda en un enlace de fibra se divide en canales ópticos (multi- canales), cada canal es una longitud de onda (λ), donde el ancho de los canales se encuentra especificada por la ITU en la norma ITU-T G.694.1 [3]. Otras



ventajas de la fibra son la baja atenuación de la señal (aproximado a los 0.2 dB/km), alta tasa de transmisión de datos, alta seguridad, baja tasa de error de bits y baja distorsión a interferencias electromagnéticas [4] [5], ver figura 1.3 y figura 1.4.



Fuente: Datos reportados por los proveedores de redes y servicios al SIUST – Colombia TIC. Fecha de consulta: 21 de marzo del 2017.

Figura 1. 2. Incremento en la suscripción a Internet fijo y móvil 4T-2015 al 4T-2016

Fuente: Tomado de [2]

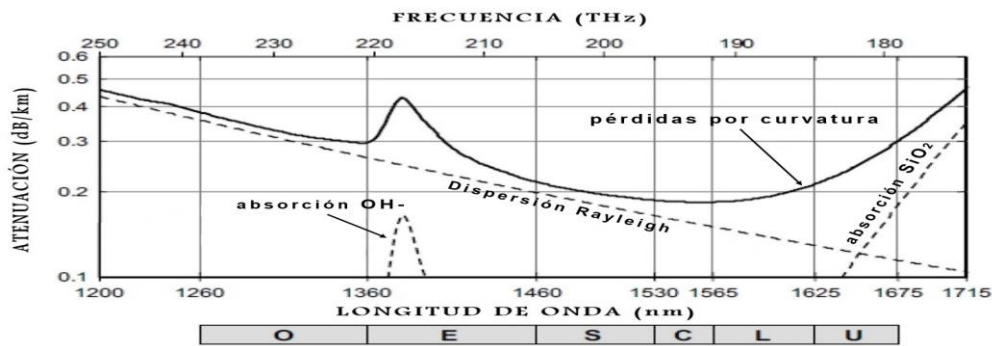


Figura 1. 3. Atenuación vs Longitud de Onda por frecuencia. Pérdidas Ópticas

Fuente: Adaptada de [3] [4].

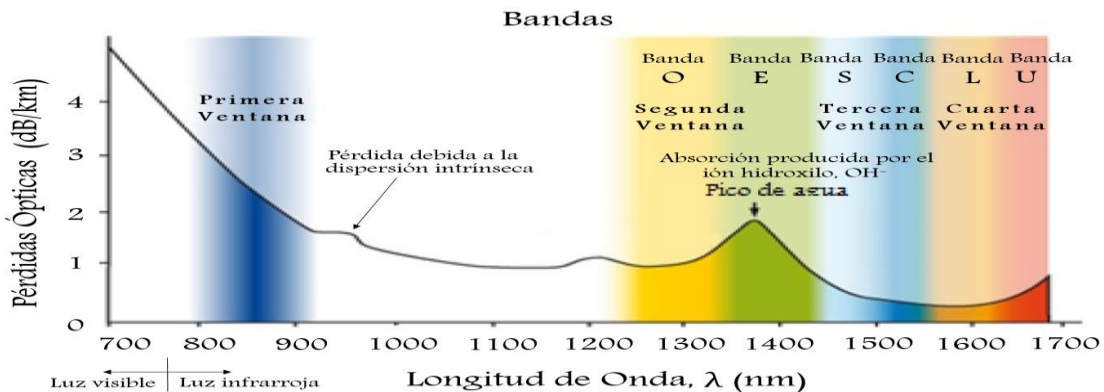


Figura 1. 4. Atenuación vs Longitud de Onda según su Ventana de Transmisión y sus Bandas

Fuente: Adaptada de [6]

Una de las estrategias que han implementado los sistemas de comunicaciones basados en fibra óptica (redes ópticas), para optimizar el uso de los recursos con los que se cuenta en una red y explotar al máximo la capacidad de la fibra, son las técnicas de multiplexación. Las técnicas de multiplexación, permiten compartir de forma eficaz un mismo medio entre diferentes sistemas de comunicaciones y principalmente minimizar el número de líneas físicas y maximizar el uso del ancho de banda de la fibra [7].

En la literatura se encuentran tres tipos de multiplexación aplicadas a redes ópticas, estas son: Multiplexación por División de Longitud de Onda (WDM, *Wavelength Divison Multiplexing*), Multiplexación por División en el Tiempo (TDM, *Time Divison Multiplexing*) y la Multiplexación por División de Espacio (SDM, *Space Divison Multiplexing*), ver [7] [8] [9]. WDM en contraste a TDM, la tasa de transmisión binaria de cada canal son más bajas, por lo que la distancia de transmisión es mucho mayor y el sistema es más escalable, es decir, la capacidad de transmisión se puede incrementar. Por otra parte, WDM en contraste a SDM, es menos costoso a la hora de ampliar el número de fibras en un enlace, para enlaces de grandes distancias el coste de infraestructura es menor ya que no requiere de un equipo de amplificación o repetidores. Es por eso, que en este trabajo de grado de maestría se utiliza la técnica de multiplexación basada en longitud de onda WDM.

El esquema general de una red óptica WDM, punto a punto, consta de un transmisor WDM, una serie de segmentos de fibra, donde el espectro del enlace de fibra contiene n longitudes de onda ($n\lambda$ s) o n –canales y un receptor WDM, ver figura 1.5. [5] [8] [9] [10]. En el transmisor, el multiplexor recibe cada una de las señales ópticas y le asigna una longitud de onda disponible en el espectro, la transmite sobre la misma fibra óptica y durante el proceso de recepción, un demultiplexor separa la señal según la información en una longitud de onda, ver figura 1.5.

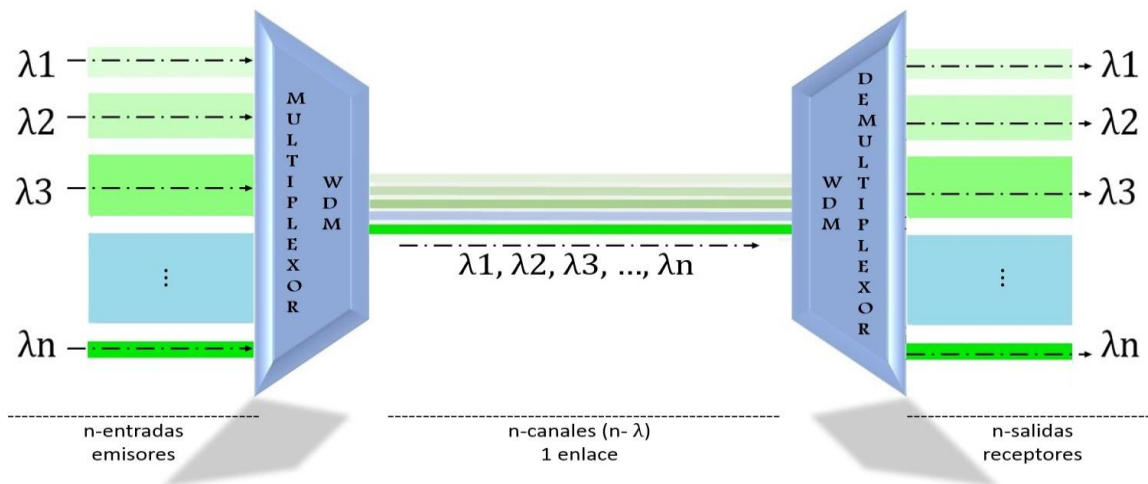


Figura 1. 5 Esquema general de una red óptica WDM

Fuente: Adaptada de [5] [8] [10].

Los sistemas WDM, se clasifican según el tipo de aplicación en: la Multiplexación por División Aproximada de Longitud de Onda –CWDM (*Coarse Wavelength División Multiplexing*) y la Multiplexación por División de Longitud de Onda Densa –DWDM (*Dense Wavelength División Multiplexing*) [5] [9] [11] [12] [13] [14]. DWDM contrasta a CWDM, según la recomendación G.694.1 de la ITU [3], trabaja con los espaciamientos típicos entre canales de 12.5GHz, 25GHz, 50GHz y 100GHz, operando en las ventanas C y L, en el rango desde los 1525nm a los 1565nm y la CWDM, según la recomendación de la ITU G.694.2 [15], opera con una separación entre canales de 2.500 GHz, ver figura 1.6. En la tabla 1.1, se presentan las principales características de la CWDM y la DWDM. Entonces, se puede afirmar que DWDM al trabajar con separaciones más densas (más pequeñas) entre canales, proporciona mayor número de canales para transmitir sobre un mismo enlace de fibra (fibra óptica) y de esta manera optimizar en gran parte el potencial de los enlaces en cuanto al ancho de banda. Es por eso, que en este trabajo de grado de maestría los enlaces de fibra se consideran WDM, utilizando la técnica de multiplexación DWDM.

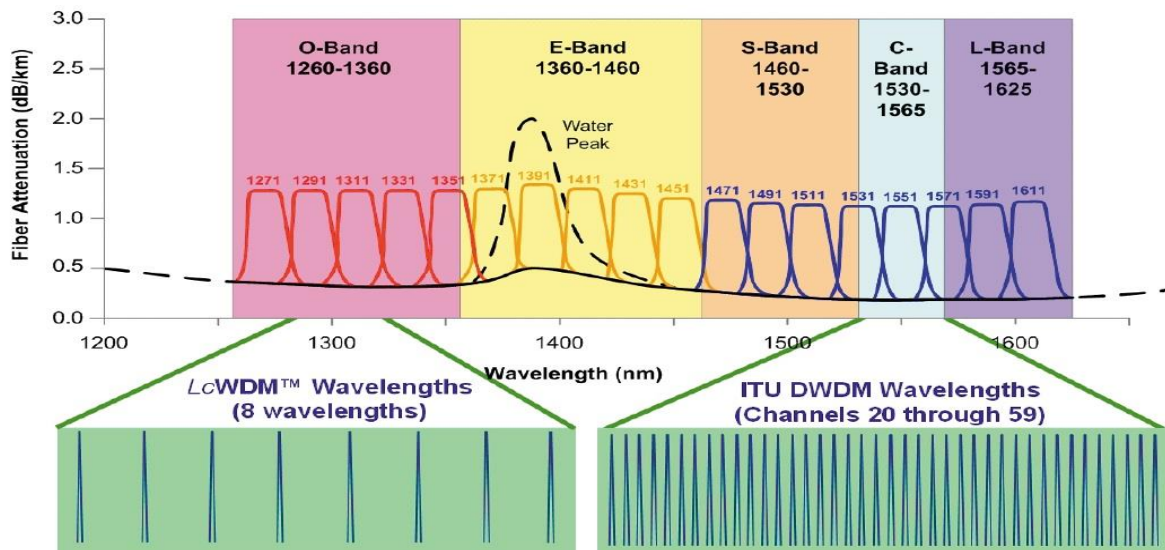


Figura 1. 6 Espaciamiento entre canales y Bandas del espectro óptico en CWDM y DWDM

Fuente: Tomado de [9]

Tabla 1. 1 Parámetros típicos en redes CWDM y DWDM

	CWDM	DWDM (MAN/WAN)	DWDM (Largo Alcance)
Espaciamiento de los Canales	20nm (2500GHz)	0.8nm (100GHz)	0.4nm (50GHz)
Ventanas	O, E, S, C, L	C, L	C, L, S
Canales por Fibra	4 -16	32 – 80	80 – 160
Capacidad por Canal	2.5Gb/s	10Gb/s	10 - 40Gb/s
Distancias Típicas	Hasta 80Km	Cientos de Km	Miles de Km
Costos	Bajo	Medio	Alto

Fuente: Adaptada de [12] [13] [14] [15].

1. 2 REDES DE CONMUTACIÓN ÓPTICA –TÉCNICAS

Un aspecto muy importante en las redes ópticas son los conmutadores ópticos (OXC, *Optical Cross-Connect*), debido a que la conmutación es la técnica que permite establecer una ruta entre dos nodos, realizando las conexiones entre un puerto de entrada (con múltiples entradas) a un puerto de salida (con múltiples salidas), ver figura 1.7. El proceso de la conmutación, se puede realizar de forma eléctrica y óptica; a nivel eléctrico, la conmutación es la causante de introducir mayor retardo en la red, debido a los procedimientos necesarios para el envío de la información y al tráfico de la red. La tabla 1.2 presenta una lista de las diferentes técnicas o formas de conmutación según la granularidad [9]. En la figura 1.7, se muestra tres de las funciones principales de un OXC: en (a) la conmutación de longitud de onda, conmuta longitudes de onda específicas desde una fibra de entrada hacia múltiples fibras de salida, utiliza dispositivos de multiplexación y demultiplexación, en (b) la conmutación de fibras, permite encaminar todas las longitudes de onda de una fibra de entrada a una de salida, y en (c) la conversión de longitud de onda, toma longitudes de onda entrantes y las convierte a otra frecuencia óptica de salida [16].

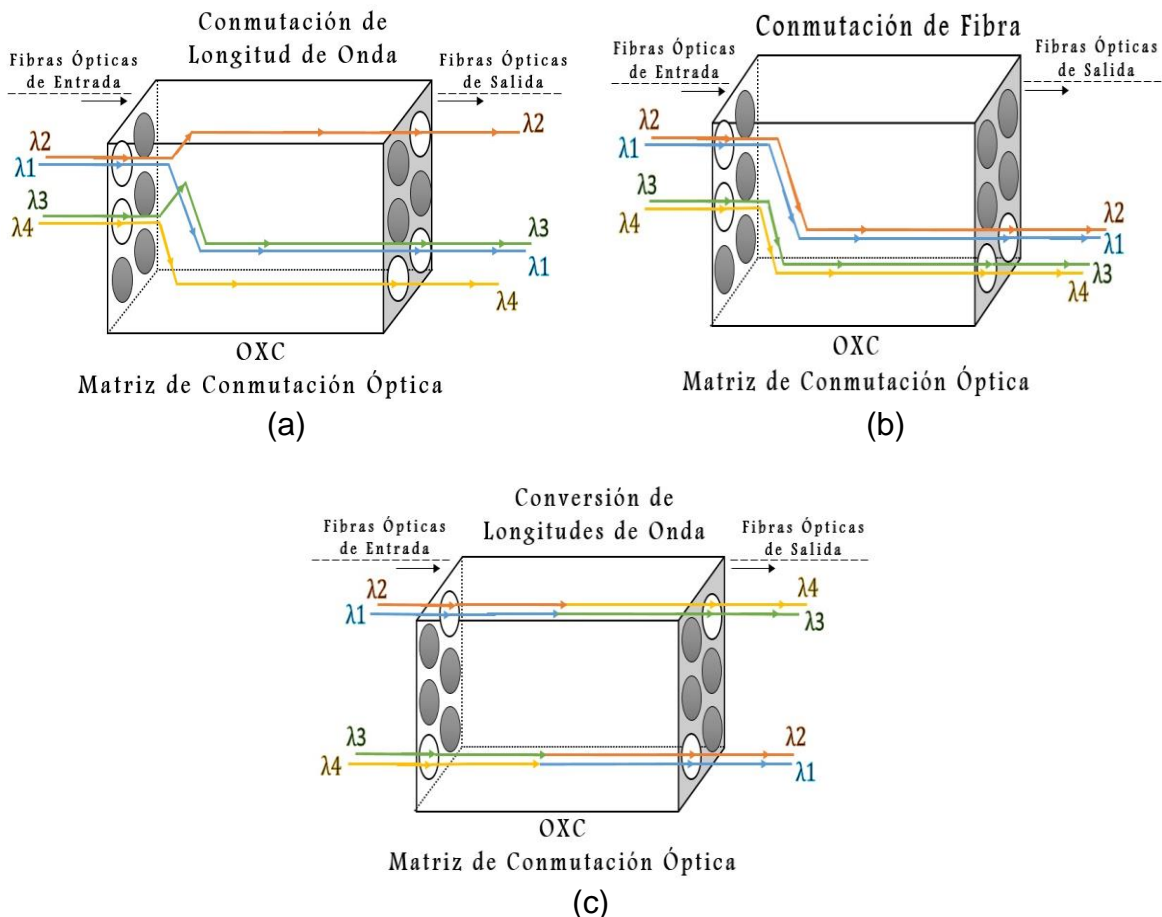


Figura 1. 7 Conversión OXC

Fuente Adaptada de [16].



En la evolución de las redes ópticas se definen tres técnicas de conmutación: Conmutación Óptica de Circuitos (OCS, *Optical Circuit Switching*), Conmutación Óptica de Paquetes (OPS, *Optical Packet Switching*) y Conmutación Óptica de Ráfagas (OBS, *Optical Burst Switching*). Estas tecnologías de conmutación basan su funcionamiento en el encaminamiento de la información a través de la red, haciendo uso eficiente de los recursos de esta, tales como ancho de banda y longitudes de onda disponibles. A continuación, se describen con más detalle, haciendo énfasis en la técnica OBS, tema central de este trabajo de grado de Maestría. En la tabla 1.3, se presenta una comparación entre la técnica de conmutación óptica OCS, OPS y OBS, señalando las principales características.

Tabla 1. 2 Clases de Granularidad en la Conmutación en redes ópticas

Clases de Granularidad en la Conmutación Óptica	
Conmutación de fibras	– Fiber Switching.
Conmutación por banda de frecuencias	– Waveband Switching.
Conmutación por longitud de onda	– <i>Wavelength Switching</i>
Conmutación por sub lambda	– <i>Subwavelength Switching</i>
Conmutación óptica de circuitos	– <i>Optical Circuit Switching</i>
Conmutación óptica de ráfagas	– <i>Optical Burst Switching</i>
Conmutación óptica de paquetes	– <i>Optical Packet Switching</i>

Fuente: Adaptada de [8] [9]

1.2 1 Conmutación Óptica de Circuitos OCS (*Optical Circuit Switching*)

La técnica de conmutación óptica OCS, se caracteriza por ser orientada a conexión, es decir, consiste en reservar un camino óptico (lightpath) entre un nodo de entrada hasta un nodo de salida, asignando una longitud de onda en cada nodo solo para ese camino (longitudes de onda dedicadas), fuera de la banda de los canales de datos y al terminar la transmisión de información se hace necesario liberar las longitudes de onda dedicadas para generar nuevos caminos ópticos, ver figura 1.8. Un lightpath, puede usar la misma longitud de onda en una ruta definida o también es posible utilizar diferentes longitudes de onda en el trayecto mediante la utilización de conversores de longitud de onda WC (*Wavelength Converter*) [17] [18] [19] [20] [21].

Algunas de las desventajas de las redes OCS son: tiene bajo nivel de escalabilidad, ya que el número de longitudes de onda que se puede transportar en una fibra es limitado, tiene un uso ineficiente del recurso ancho de banda, ya que las longitudes de onda en una fibra son dedicadas a un lightpath específico, Además, toda la información que es transmitida se somete a conversiones

eléctrico-óptico (EO, *Electronic-Optical*) y óptico-eléctrico (OE, *Optical-Electronic*) en los nodos de ingreso y destino, lo que incrementa los retardos en la red.

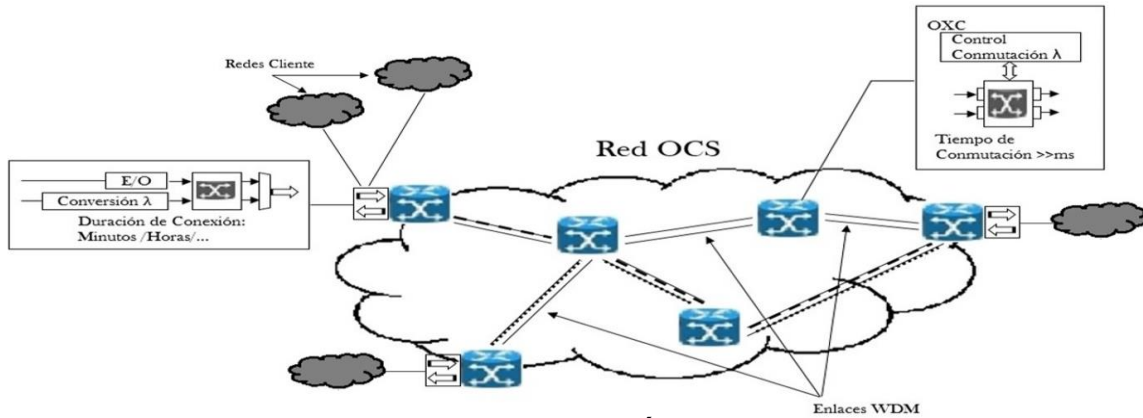


Figura 1. 8 Red de Conmutación Óptica de Circuitos OCS

Fuente: Adaptada de [9] [17] [20].

1.2 2 Conmutación Óptica de Paquetes OPS (*Optical Packet Switching*)

La técnica de Conmutación Óptica OPS, emplea un formato de pequeños bloques o unidades de datos denominados paquetes. Cada paquete se compone por un encabezamiento (paquete de control) y la carga útil (Payload). En las redes OPS, la transmisión consiste en la conversión inicial de los paquetes electrónicos al dominio óptico, para ser transportados por la red sin ningún tipo de conversión entre los extremos, ver figura 1.9. Donde, la información en el nodo origen es segmentada en paquetes, cada paquete se envía junto con su cabecera, el cual, transporta toda la información de configuración para el proceso de conmutación en cada nodo. OPS, proporciona mayor flexibilidad y tratamiento de los recursos ópticos, puesto que los canales para el transporte de la información y el establecimiento se hacen de manera dinámica [17] [18] [19] [20] [21] [22].

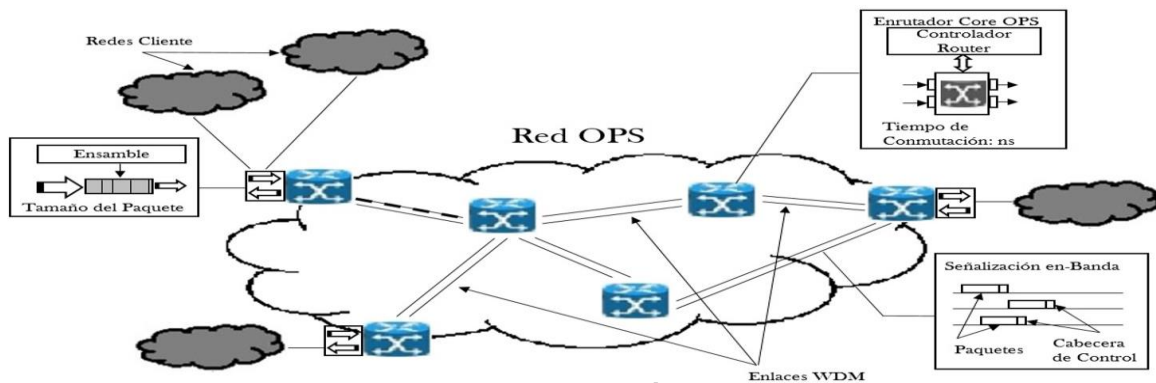


Figura 1. 9 Red de Conmutación Óptica de Paquetes OPS

Fuente: Adaptada de [9] [17] [21].



Algunas de las desventajas de las redes OPS, son: actualmente los conmutadores no pueden realizar todavía un procesamiento óptico, por lo que es necesario pasar la cabecera al dominio eléctrico para realizar el encaminamiento. Además, actualmente las redes OPS no cuenta con un sistema de almacenamiento óptico, dado que los paquetes al llegar a un nodo de red deben esperar ya sea porque su cabecera está procesando o porque en ese momento otro paquete está usando esa misma interfaz de salida. Por último, está la sincronización del paquete de información con su cabecera, lo que aumenta la complejidad de implementación.

Tabla 1. 3 Características principales de la Conmutación Óptica OCS, OPS y OBS

	OCS	OBS	OPS
MODELO DE CONMUTACIÓN	Conmutación de Circuitos.	Multiplexación estadística (de ráfagas).	Multiplexación estadística (de paquetes).
UTILIZACIÓN ANCHO DE BANDA	Baja.	Alta.	Alta.
ESQUEMA DE SEÑALIZACIÓN	Dos caminos, Fuera de banda.	Un camino. Fuera/dentro de banda.	Un camino. Dentro de banda.
TIEMPO DE CONMUTACIÓN	>> ms.	μs.	ns.
RETARDO DE LA RED	Muy bajo.	Depende del esquema de Ensamble de ráfagas.	El retardo es bajo, incluso con Líneas de Retardo de Fibra (FDLs).
BUFFER ÓPTICO	No Requiere.	No Requiere.	Requiere.
COMPLEJIDAD DE ALGORITMOS DE CONTROL EN EL NODO	Muy baja.	Muy alta: -Problemas de resolución de contenciones. -Problema de ensamblado Ráfagas/Paquete. -Priorización de Ráfagas/Paquetes.	
DURACIÓN DE CONEXIÓN EN UN NODO	Muy larga.	Depende del esquema de reservación de recursos.	Duración del paquete.

Fuente: Adaptada de [9] [18] [20] [22]

1.2 3 Conmutación Óptica de Ráfagas OBS (Optical Burst Switching)

La técnica de conmutación óptica por ráfagas OBS, tiene su origen en el año 1999 con Qiao y Yoo, ver [23], se coloca en la evolución de las redes de alta velocidad, en comparación a la OCS y la OPS, como la técnica más favorable para las redes ópticas actuales y como base para las siguientes generaciones. La OBS, es un tipo de híbrido entre OCS y OPS, debido a que, combina lo mejor de la conmutación de paquetes y de la conmutación de circuitos, a la vez que minimiza las desventajas de las mismas. [17] [18] [19] [20] [21].

Algunos problemas que las redes OBS logran superar, presentes en las redes OCS, es la utilización de ancho de banda, latencia de configuración y

adaptabilidad del volumen tráfico. Además, supera a las redes OPS, en cuanto a que no necesita de un sistema de almacenamiento óptico, es por eso, que actualmente no es posible la implementación de las redes OPS en su totalidad. Por tratarse del tema de investigación de este trabajo de grado de Maestría, a este ítem se dedicará la sección siguiente, estudiando su arquitectura, esquema funcional y se hace énfasis principalmente en el proceso de ensamble de ráfagas.

1. 3 REDES DE CONMUTACIÓN ÓPTICA DE RÁFAGAS

Las redes ópticas por Multiplexación por División de Longitud de Onda, redes WDM, adoptan la técnica de conmutación óptica por ráfagas, redes OBS, para transmitir de una manera más eficiente la información, destacando como estrategia principal la granularidad en la conmutación debido a la multiplexación estadística, por tanto, logran soportar en cierto modo las altas demandas sobre las redes actuales, que presentan un incremento del volumen del flujo de tráfico entrante con requerimientos más exigentes en cuanto a tiempo real y en ancho de banda [23]. En las redes OBS, la unidad de transporte es la ráfaga, la cual contiene un número determinado de paquetes de datos encapsulados (ensamblados) y se le asigna un encabezamiento o paquete de control, llamado BCP (*Burst Control Packet*), con información específica de la ráfaga. El BCP, es un paquete aparte de la ráfaga y se transmite por un canal dedicado diferente al canal de transmisión o longitud de onda (λ) de la ráfaga, ver figura 1.10, [9] [20]. Donde, los BCPs le informan a la red el flujo de información que transita sobre ella en forma de ráfaga para que esta pueda conmutar la ráfaga hacia el nodo borde de salida. Entonces, las redes OBS contienen múltiples paquetes que se conmutan en un medio completamente óptico y se transmiten en forma de ráfagas.

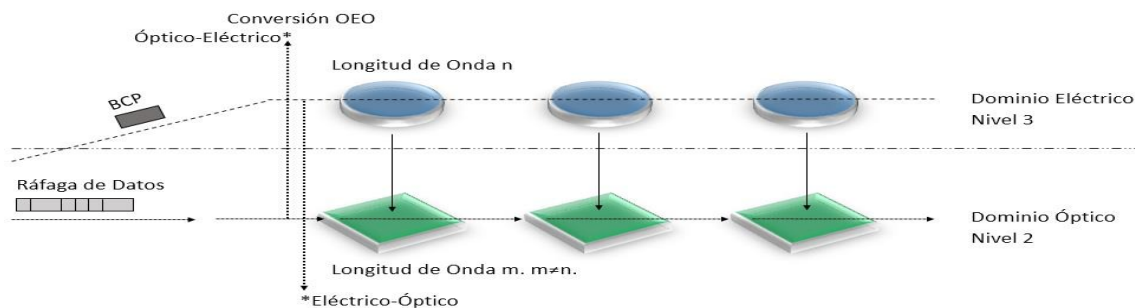


Figura 1. 10 Esquema de transmisión del BCP y de la Ráfaga

Fuente: Adaptada de [9] [20] [21] [23] [24]

El BCP y la ráfaga se transmiten en el dominio óptico, separados físicamente por un intervalo de tiempo denominado offset o tiempo de compensación y en diferentes canales o longitudes de onda, ver el esquema de transmisión en la figura 1.10. En cada uno de los nodos, la ráfaga se transmite sin procesamiento alguno, mientras que, para el BCP se debe realizar conversión de señales óptico-eléctrico-óptico OEO (*Optical Electronic Optical*), la cual, aplica la conversión OE, del medio óptico al medio eléctrico, para su procesamiento y para su transmisión

aplica la conversión EO, del medio eléctrico al medio óptico, [9] [18] [21] [24]. En la figura 1.11 se observa los tiempos para el envío del BCP y de la ráfaga.

Las ráfagas no esperan a la confirmación del establecimiento de los canales o longitudes de onda en cada uno de los nodos intermedios o nodos centrales, esta comienza a transmitir la información una vez haya transcurrido el tiempo de offset, el cual, especifica la duración de la ráfaga e informa el momento en el que el nodo debe reconfigurarse para la recepción de una nueva ráfaga. El tiempo de offset se calcula mediante la suma de los tiempos de conmutación de cada nodo intermedio, donde se tienen en cuenta las conversiones OEO, el procesamiento de los BCP, el tiempo de respuesta de la matriz de conmutación, el análisis de la distancia más corta en el enlace extremo a extremo y el factor velocidad de propagación de cada longitud de onda.

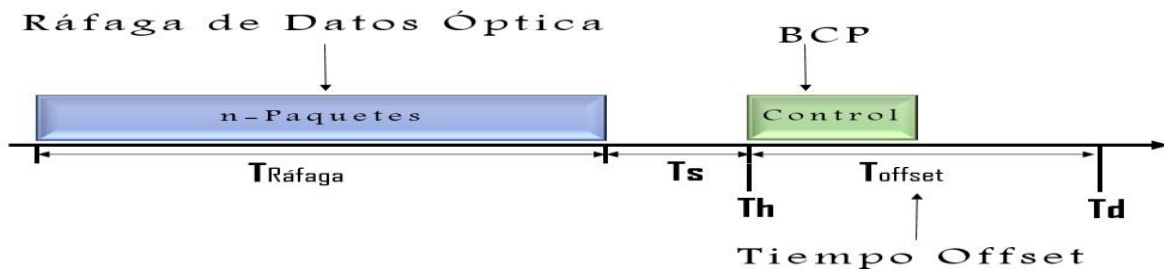


Figura 1. 11 Diagrama de tiempo para el envío de la ráfaga y de su BCP

Fuente: Adaptada de [9] [19] [20]

Donde,

$T_{Ráfaga}$: Tiempo de ensamble de la ráfaga.

T_s : Tiempo de planificación.

T_h : Tiempo de envío del BCP.

T_{offset} : Tiempo de Offset.

T_d : Tiempo de envío de la ráfaga de datos.

1.3. 1 Arquitectura y Esquema Funcional de una Red OBS

La arquitectura general de las redes troncales basadas en la conmutación óptica por ráfagas, redes OBS, al igual que toda red de comunicaciones, se modela bajo la estructura base, un arreglo de nodos y enlaces, ver figura 1.12. [8] [9]. Los nodos se clasifican en nodo frontera o borde (Edge Node) y en nodo central o núcleo (Core Node) y a su vez, el nodo borde según su función puede ser de entrada o de ingreso (edge node ingress) y de salida (edge nodo egress), ver figura 1.13. Los enlaces, se encargan de enviar la información a través de diferentes medios de transmisión y de la interconexión entre nodos. La arquitectura de una red OBS se diferencia entre las otras redes ópticas, principalmente en el nodo borde, nodo borde de entrada y nodo borde de salida, en el nodo borde de entrada se realiza el proceso de transmisión y ensamble de las ráfagas mientras que en los nodos borde de salida ocurre el desensamblaje de

los datos, proceso contrario al ensamble de la ráfaga. Por otro lado, los nodos centrales realizan el proceso de conmutación óptica mediante multiplexores y demultiplexores y con una unidad de control, la cual procesa los paquetes de control (BCP) de la ráfaga.

El modelo de transmisión dentro de la arquitectura de una red OBS se basa en su esquema funcional, este clasifica los nodos OBS según sus procesos internos, donde cada proceso cumple con una función de vital importancia para el buen desempeño total de la red, ver figura 1.12 y figura 1.13. El modelo general de transmisión OBS, funciona: los paquetes generados por los usuarios, redes de usuarios origen, llegan a un nodo borde de entrada donde son ensamblados en ráfagas utilizando un algoritmo de ensamble, una ráfaga es una agrupación de datos o paquetes que se dirigen hacia un mismo destino con características de prioridad y calidad definida, donde el envío de los BCPs precede a la transmisión de cada ráfaga, estos son enviados a través del núcleo de la red, el cual, está compuesto por nodos con conmutadores OXC, cuya función principal es encaminar la ráfaga hasta un nodo borde de salida, en este punto, las ráfagas se desensamblan y los paquetes son entregados a los usuarios finales, a las redes de usuarios destino, ver figura 1.12 y figura 1.14, [9] [10] [18] [20].

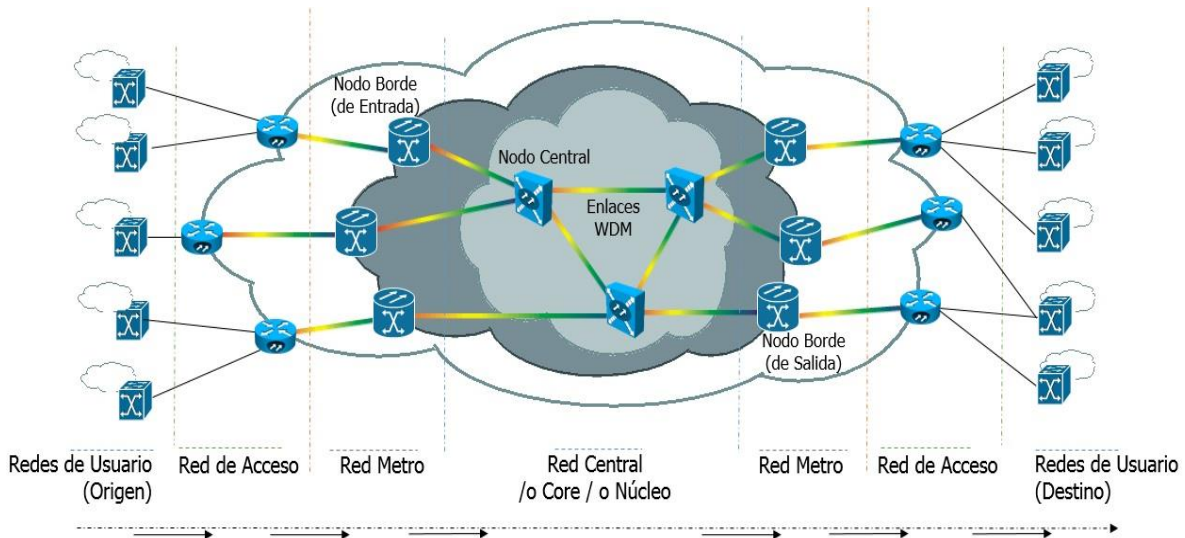
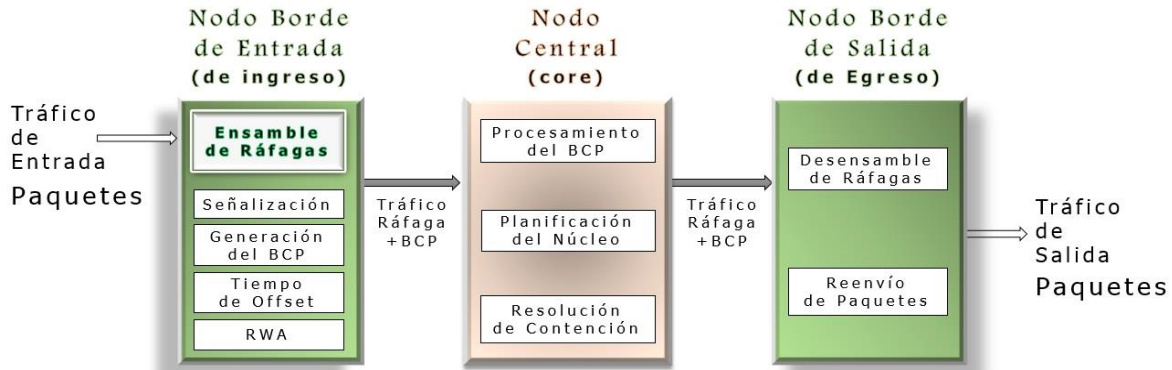


Figura 1. 12 Arquitectura general de una red de comunicación óptica

En la figura 1.13 se presenta el esquema funcional de una red OBS, el cual, clasifica de la siguiente manera cada uno de los nodos OBS según sus procesos internos: los procesos del nodo borde de entrada constan del ensamble de ráfagas, el enrutamiento, asignación de longitud de onda y la sincronización de la ráfaga en la frontera de la red. Por otro lado, los procesos del nodo central constan de la señalización, conmutación de las ráfagas y la resolución de contiendas. Finalmente, los procesos del nodo borde de salida constan del desensamble de las ráfagas y el envío de los paquetes a las capas superiores.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA



RWA: Encaminamiento y asignación de longitudes de onda (Routing and Wavelength Assignment)

BCP: Paquete de Control de Ráfaga (Burst Control Packet)

Figura 1. 13 Clasificación de los Nodos de Red según el Esquema funcional de una red OBS

Fuente: Adaptada de [9] [17] [20]

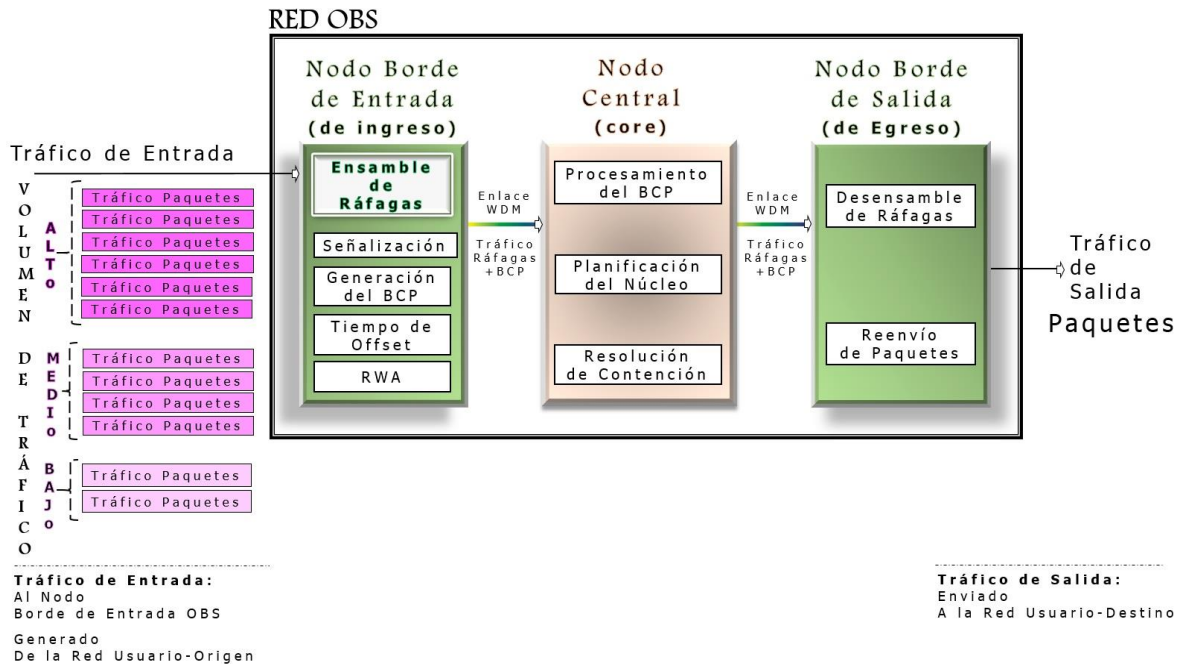


Figura 1. 14 Modelo de Transmisión según el Esquema funcional en una red OBS

1.3.1.1. Nodo de Borde

El nodo de borde OBS tiene dos componentes: nodo de entrada y nodo de salida, el nodo borde de entrada, es el encargado del inicio de la transmisión, el cual, recibe, soporta y administra el volumen del tráfico que ingresa a la red, transmitiendo los paquetes (información) hacia los nodos centrales en un formato llamado ráfaga (de tamaño aleatorio) y su paquete de control BCP, ver figura 1.14, figura 1.15. El tráfico que ingresa a la red puede provenir de diferentes redes,

como: Protocolo Internet (IP, *Internet Protocol*), Red Óptica Síncrona (SONET, *Synchronous Optical NETWORK*), Gb-Ethernet, Modo de Transferencia Asíncrona (ATM, *Asynchronous Transfer Mode*), entre otros [9] [17] [21] [25] [26]. ver figura 1.15.

La arquitectura de un nodo borde de entrada se muestra en la figura 1.16, la cual, presenta los procesos desde que el paquete llega al nodo, clasifica cada paquete según la dirección de destino o tipo de servicio o prioridad, entre otros, o la combinación entre ellos, agrupa cada paquete en la cola o Buffer correspondiente, ensambla la ráfaga, genera el BCP y calcula el tiempo de offset. El BCP se envía por el canal de control predefinido y la ráfaga, según el criterio seleccionado, se envía por uno de los canales de datos disponibles de la fibra hacia el nodo borde destino, dicho criterio puede ser el canal menos usado, ver figura 1.15 y figura 1.16, [9] [17] [19] [20] [21] [24].

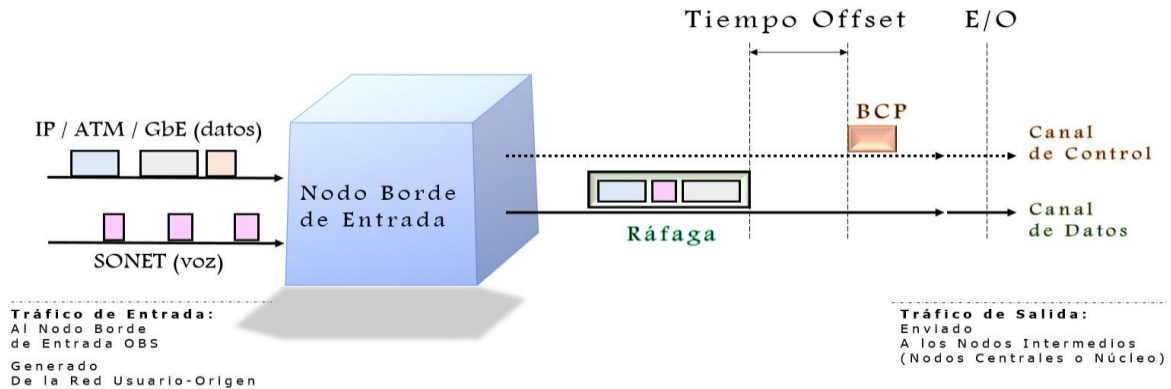


Figura 1. 15 Ensamble de Ráfagas en un Nodo Borde de Entrada

Fuente: Adaptada de [9] [18] [21] [25]

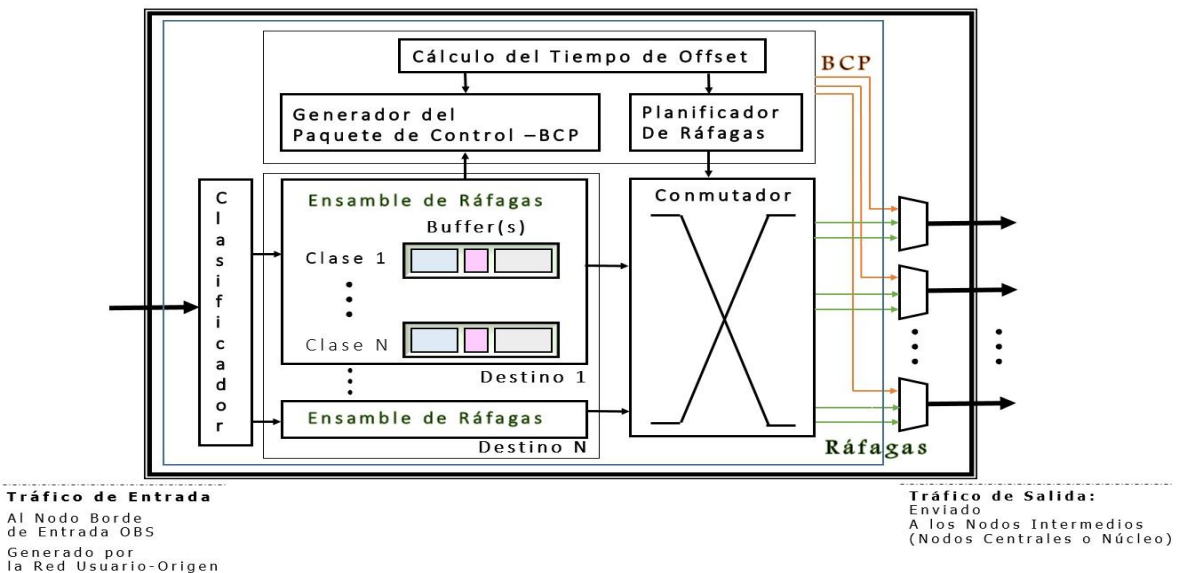


Figura 1. 16 Arquitectura de un Nodo Borde de Entrada OBS

Fuente: Adaptada de [9] [17] [19] [20] [21] [24] [26]

El nodo borde de salida, en una red OBS, tiene como función principal el desensamble de las ráfagas, donde la ráfaga es separada en paquetes y cada paquete es transmitido a la red usuario-destino correspondiente, mediante la función de reenvío de paquetes, ver figura 1.17. Internamente su arquitectura funcional incluye múltiples receptores de ráfagas, módulos para desensambles de la ráfagas, planificadores y gestores de colas de paquetes, un conmutador y múltiples interfaces de redes de usuario destino, ver figura 1.14 y figura 1.17.

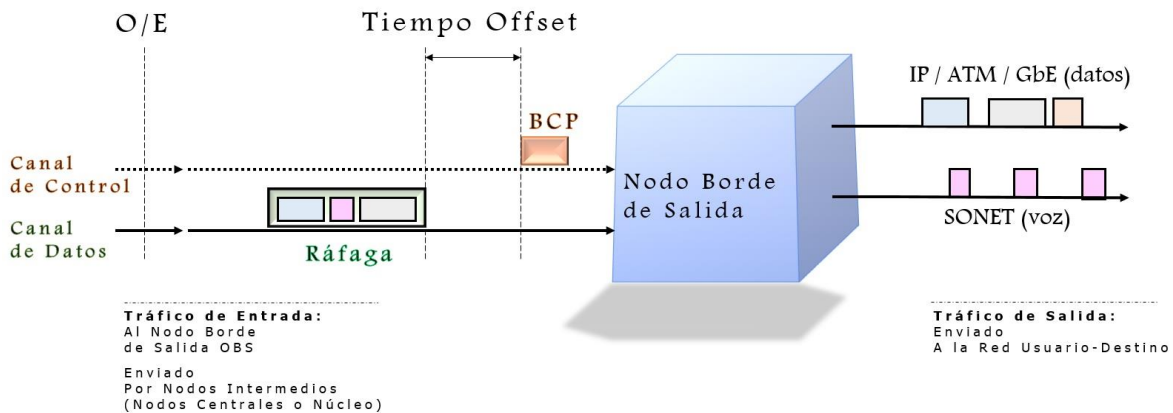


Figura 1. 17 Desensamble de Ráfaga en un Nodo Borde de Salida

Fuente: Adaptada de [9] [21] [25]

1.3.1.1.1. Ensamble de Ráfagas

El ensamble de ráfagas es el primer proceso de una red OBS, ocurre en el nodo borde de entrada, ver figura 1.14 y figura 1.16, el cual agrupa varios paquetes, provenientes de las capas superiores (por ejemplo tráfico IP), dentro de un macro-paquete llamado ráfaga, figura 1.15, envía éste a un determinado destino como si se tratara de un único paquete y a su vez, genera y envía el paquete de control (cabecera) de cada ráfaga, teniendo en cuenta un tiempo offset. Cabe resaltar dos ventajas del proceso de ensamble: 1) principalmente, administra el volumen del flujo de tráfico entrante a la red; 2) reduce el número de conversiones OEO (Óptica-Eléctrica-Óptica) en los conmutadores de los nodos centrales, ya que, la ráfaga o carga útil se transmite sin conversión, 100% por el medio óptico, por otro lado, para cada BCP analizado en un nodo le corresponde una conversión por cantidad de paquetes.

Por ser el tema central de investigación de este trabajo de grado de Maestría, se describen con más detalle los módulos del proceso de ensamble, algunos de los algoritmos de ensamble existentes en la literatura y su clasificación, en una sección más adelante.

1.3.1.1.2. Señalización

La señalización es el proceso de configuración de mecanismos de conexión, el cual, establece la comunicación en una red OBS entre dos nodos borde a través de nodos centrales. La función principal de este proceso, está en determinar el funcionamiento de la red y además en asignar los recursos de red requeridos para la transmisión de la ráfaga.

Los dos tipos de señalización son: distribuida y centralizada. En la primera, la reserva se realiza nodo a nodo y en la segunda es un elemento central con conocimiento de toda la red, se encarga de enviar las solicitudes de reservas de recursos a los nodos involucrados [9] [21] [24] [26] [27]. Por lo general, la mayoría de los trabajos implementan el primer tipo de señalización, señalización distribuida, basadas en el principio TAG (*Tell And Go*), ver figura 1.18. En este trabajo de grado de Maestría se usa la señalización distribuida.

Dependiendo de la forma de reservar las rutas para una ráfaga un esquema de señalización se puede caracterizar por:

- Reservación iniciada en la fuente, en el destino o en un nodo intermedio.
- Reservación inmediata o retardada.
- Reservación en una vía (one way), en dos vías (two way) o híbrido.
- Reservación persistente o no persistente.
- Señalización centralizada o distribuida.
- Liberación de recursos implícita o explícita.

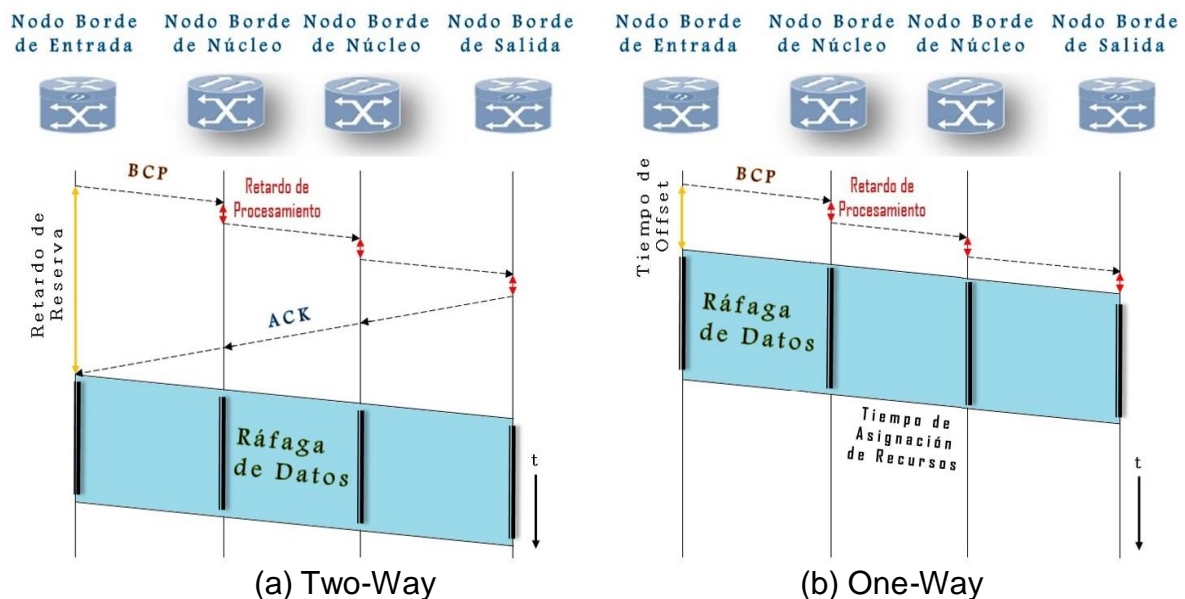


Figura 1. 18 Tipos de Señalización OBS (a) centralizada con two-way y (b) distribuida con one-way

Fuente: Adaptada de [9] [21] [24] [27]

En las redes OBS, existen dos protocolos para la señalización: (1) el protocolo JIT (*Just In Time*), un nodo OBS configura los conmutadores ópticos para la ráfaga, tan pronto como es recibido y procesado el paquete de control correspondiente, los recursos del nodo se ponen a disposición antes de la llegada de la ráfaga sin tener en cuenta tiempos de offset, en la mayoría de los casos esto llevaría al uso ineficiente de la red; (2) el protocolo JET (*Just Enough Time*), es el más utilizado, debido a que muestra una mejor utilización del ancho de banda al realizar la reserva de recursos únicamente desde la llegada de la ráfaga hasta que es conmutada y no desde la llegada del paquete de control al nodo, ver figura 1.19. [9] [17] [19] [20] [24] [26] [27].

Por lo anterior, para este trabajo de grado de Maestría, para la señalización en la red OBS de prueba sea escogido el protocolo JET. Donde, el tiempo de asignación y liberación de recursos es estimado y solo se asigna el tiempo correspondiente a la duración de la ráfaga, el cual, depende de la cantidad de paquetes que conforma la ráfaga.

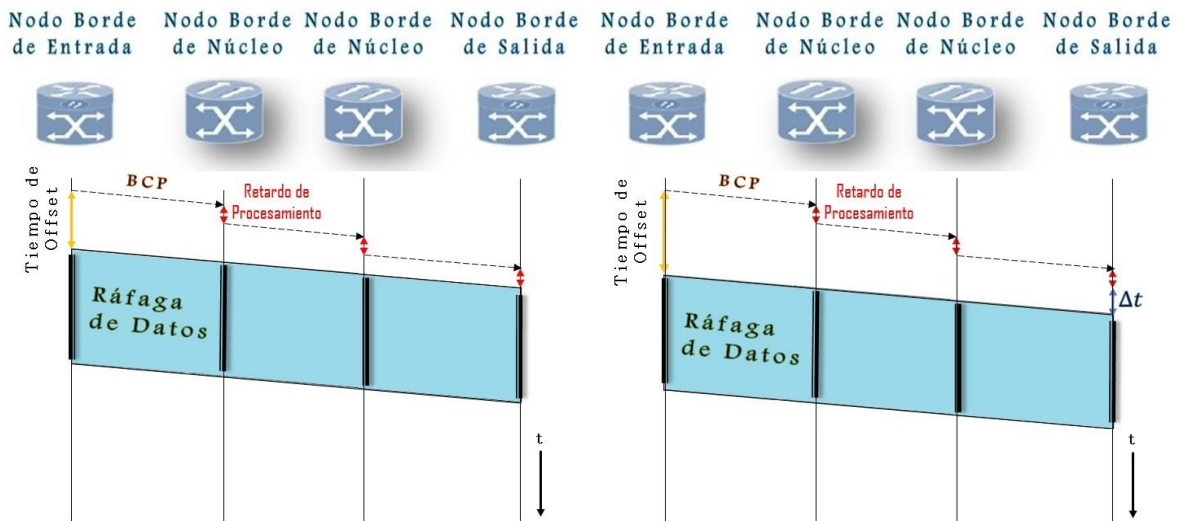


Figura 1. 19 Protocolo de Señalización JET

Fuente: Adaptada de [9] [17] [18] [20] [21] [26] [27]

1.3.1.1.3. Enrutamiento y Asignación de Longitud de Onda

El proceso de enrutamiento (*routing*) en una red de comunicación óptica, consiste en buscar el conjunto de enlaces y nodos, creando un camino adecuado, desde el nodo origen al nodo destino, con el fin de encaminar la información. Por lo tanto, el enrutamiento está relacionado directamente al proceso de reserva del canal de transmisión o longitud de onda, en cada nodo [9] [19] [21] [26] [28]. El proceso de asignar una longitud de onda adecuada por la cual se va a transportar la información, se conoce como asignación de longitud de onda (WA, *Wavelength*



Assignment). La unión de los dos procesos se les conoce como RWA (*Routing and Wavelength Assignment*).

Los algoritmos aplicados a RWA, realizan de manera simultánea los dos procesos, determinar el camino adecuado entre dos nodos de la red (desde nodo origen al nodo destino) y asignar una longitud de onda adecuada para el transporte de la información, sin embargo, ello implica mayor complejidad computacional. Pero, es posible resolver los dos problemas de manera simultánea (R + WA), reduciendo la complejidad computacional.

Las estrategias más utilizadas para realizar el proceso de enrutamiento en las redes OBS son: enrutamiento fijo, enrutamiento alternado y enrutamiento adaptativo [28].

- **Enrutamiento fijo:**

Consiste en elegir la ruta más directa entre los nodos, esta conexión origen-destino siempre es la misma (ruta fija), si dicha ruta llegase a fallar o no es posible establecer el camino óptico se rechaza la petición. Esta técnica busca disminuir el tiempo de establecimiento de conexión, pero es propensa a tener alta probabilidad de bloqueo. Se destacan las siguientes técnicas de enrutamiento: algoritmo de camino más corto (SP, *Shortest Path*), algoritmo peso por salto (HW, *Hop Weight*).

- **Enrutamiento fijo alternado:**

Establece un determinado número de caminos ópticos calculados a través de una función de costo fija, si el primer camino no se logra establecer se verifica con los otros caminos, si no se logra establecer ruta, la solicitud de conexión es rechazada. Con esta técnica se obtiene una mejora en la capacidad de la red y en la probabilidad de bloqueo, aunque con un tiempo de conexión más largo debido a número de rutas que se calcula.

- **Enrutamiento adaptativo:**

El establecimiento de las rutas se lleva a cabo en tiempo real, es decir, las conexiones se actualizan al establecer o terminar una conexión. Esta técnica trabaja creando enlaces dinámicamente, debido a esto se requiere un tiempo mayor para generar el camino óptico, pero permite disminuir la probabilidad de bloqueo de la red.

Las técnicas para la asignación de longitud de onda se efectúan de tres maneras: (1) organizando las longitudes de onda de acuerdo al orden en el que fueron usadas en la última ocasión; (2) realizando una elección aleatoria; (3) haciendo uso de expansión o contracción del espectro de las longitudes de onda que han sido usadas en los enlaces. A continuación, se describirán algunos de los algoritmos de asignación de longitud de onda [28]:



- **Método Aleatorio** (R, *Random*)

Luego de haber establecido una ruta, el algoritmo heurístico aleatorio hace una búsqueda entre las longitudes de onda disponibles y selecciona una de ellas aleatoriamente.

- **Método de Primer Ajuste** (FF, *First Fit*)

Se realiza una numeración de las longitudes de onda existentes, cuando se desea realizar una conexión se elige entre dichas longitudes la primera que se encuentre disponible empezando desde la de menor numeración hasta la de mayor numeración.

- **Método del Menos Usado** (LU, *Least Used*)

Selecciona la longitud de onda que se esté usando menos en la red, con el objetivo de hacer un balance de cargas en todas las longitudes de onda.

- **Método del Más Usado** (MU, *Most Used*)

Es un método heurístico, el cual selecciona la longitud de onda que más se esté usando en la red, ofrece un mejor desempeño que los métodos “menos usado” y “primer ajuste”.

Para este trabajo de grado de Maestría, para la red OBS de prueba, se utilizará el algoritmo de enrutamiento de caminos más cortos entre los nodos origen y destino (número de saltos), el algoritmo Dijkstra, también llamado algoritmo de caminos mínimos. De igual manera, se implementa el algoritmo LAUC (Último Canal Libre Disponible) para el proceso de asignación de longitud de onda [9] [26].

1.3.1.2. Nodo Central

El nodo central o núcleo en una red OBS, se encarga principalmente de la configuración de su matriz de conmutación para que la ráfaga salga por el puerto adecuado y transmitirla sobre la fibra óptica, esto en el caso de una red distribuida. Todo este proceso está dado en función de tres elementos: (1) un OXC (*Conmutador y Matriz de Conmutación*); (2) una unidad de control electrónica (CU, *Control Unit*); (3) por convertidores óptico-electrónico OE (*Optical Electronic*) y electrónico-óptico EO (*Electronic Optical*). El OXC, conmuta las señales con una longitud de onda de salida, el cual puede ser con independencia de la longitud de onda a la entrada, cuyo objetivo es ofrecer mayor capacidad de transmisión y la flexibilidad que requiere el tráfico de las redes actuales. La CU, se encarga de modificar automáticamente la matriz al realizar el proceso de planificación de las ráfagas de datos debido a que gestiona la información de control, comprobando que los recursos solicitados estén o no disponibles. Por último, el convertidor óptico-electrónico OE y el electrónico-óptico EO, son los encargados de procesar las señales en cada nodo y de esta manera analizar en el medio eléctrico la

información de los BCPs para la configuración de las OXC y realizar el encaminamiento de las ráfagas en el medio óptico [9] [26].

Los procesos principales del nodo core en una red OBS, son:

- **Procesamiento del BCP:**

En cada uno de los nodos núcleo OBS se realiza una separación entre las señales de datos y de control (o BCP), donde el procesamiento del BCP se lleva a cabo en la Unidad de Control, ver figura 1.10 y figura 1.20, donde analiza la información del BCP para administrar la matriz de conmutación óptica OXC, información como la longitud de onda asignada a la ráfaga, la longitud de onda dedicada para el BCP, el tiempo de offset, la calidad de servicio, el tamaño de la ráfaga, otros [9] [21] [24] [27].

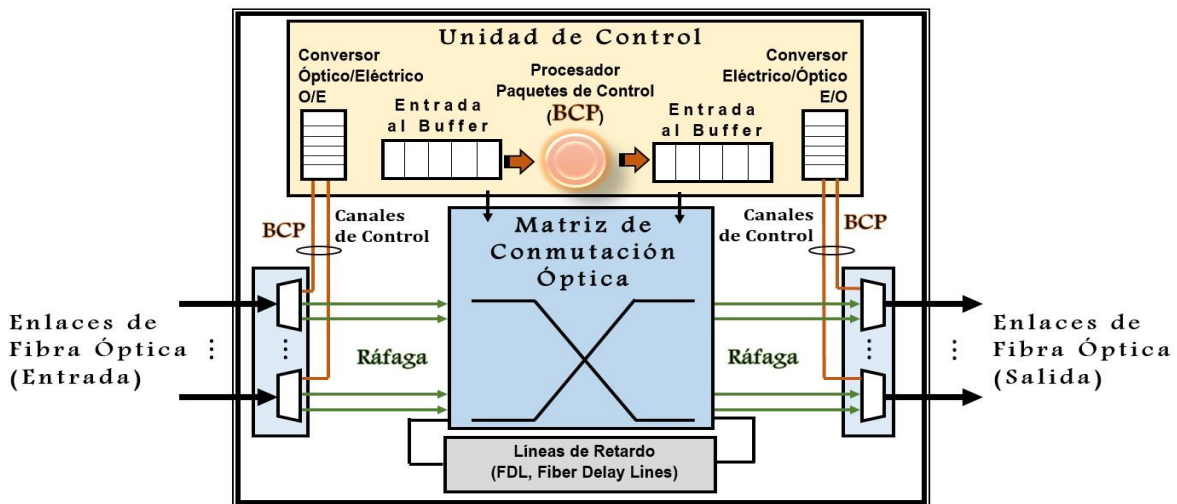


Figura 1. 20 Arquitectura de un Nodo Central OBS

Fuente: Adaptada de [9] [17] [18] [20] [24] [26]

- **Resolución de Contención**

La contención se presenta cuando dos o más ráfagas requieren al mismo tiempo un mismo recurso o cuando una ráfaga llega a un nodo OBS y todos los recursos están ocupados. Algunas de las técnicas de resolución de contención son: (1) líneas de retardo de fibra FDL (*Fiber Delay Lines*), se utilizan cuando dos o más ráfagas contienden en un nodo. El objetivo es retrasar las ráfagas y enviarlas secuencialmente a las salidas. Es un mecanismo aceptado, aunque son costosas y se requieren grandes extensiones físicas de fibra; (2) deflexión de rutas, conocida también como el método hot potatoe, el cual, consiste en que una de las ráfagas utiliza el puerto de salida causante de la congestión y las demás se envían por rutas alternativas; (4) segmentación de ráfaga, consiste en dividir la ráfaga en segmentos y encaminarlos por diferentes rutas. [9] [17] [19] [20] [21] [24] [27].

- Planificación

Los nodos OBS planifican los recursos en los conmutadores ópticos analizando la información de cada BCP o paquetes de control, de tal manera que las ráfagas puedan pasar a través de ellos sin procesamiento y sin conversión OEO [9] [19] [20] [21] [24]. Los esquemas de planificación de recursos se clasifican según los tiempos de comienzo y fin de ocupación de los mismos, de la siguiente manera:

- **Establecimiento explícito:** la estrategia consiste en que se reserva una longitud de onda y se configura el conmutador después de recibir y procesar el BCP.
- **Establecimiento estimado:** la estrategia consiste en que la reserva de una longitud de onda se retarda y se utiliza la información del BCP para estimar el tiempo de arribo de la ráfaga correspondiente en el nodo OBS.
- **Liberación explícita:** la estrategia consiste en que los nodos de borde se encargan de enviar un tráiler adicional al BCP para indicar el fin de la ráfaga, después de esto el nodo OBS libera la longitud de onda.
- **Liberación estimada:** la estrategia consiste en que el nodo estima el fin de la ráfaga, utilizando la información del tiempo de offset y el tamaño de la información transportada en el BCP correspondiente.

Por último, para esta sección, se muestra la diferencia de la conmutación de ráfagas en un nodo borde de entrada OBS y en un nodo core OBS en la figura 1.21 y en la figura 1.22, respectivamente. Donde, el primero se encarga de generar la ráfaga mediante un mecanismo o algoritmo de ensamble con su BCP (paquete de control) enviando cada uno por su longitud de onda asignada y el otro, se encarga de transmitir la ráfaga y procesar su BCP.

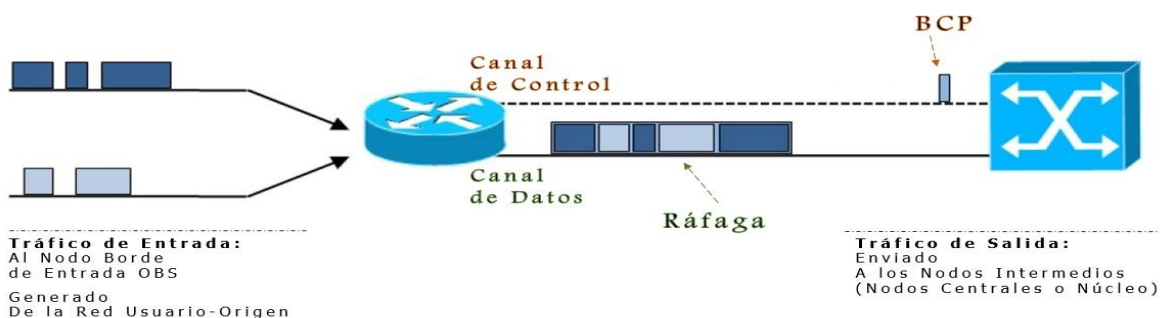


Figura 1. 21 Conmutación de Ráfagas en un Nodo Borde de Entrada

Fuente: Adaptada de [27]

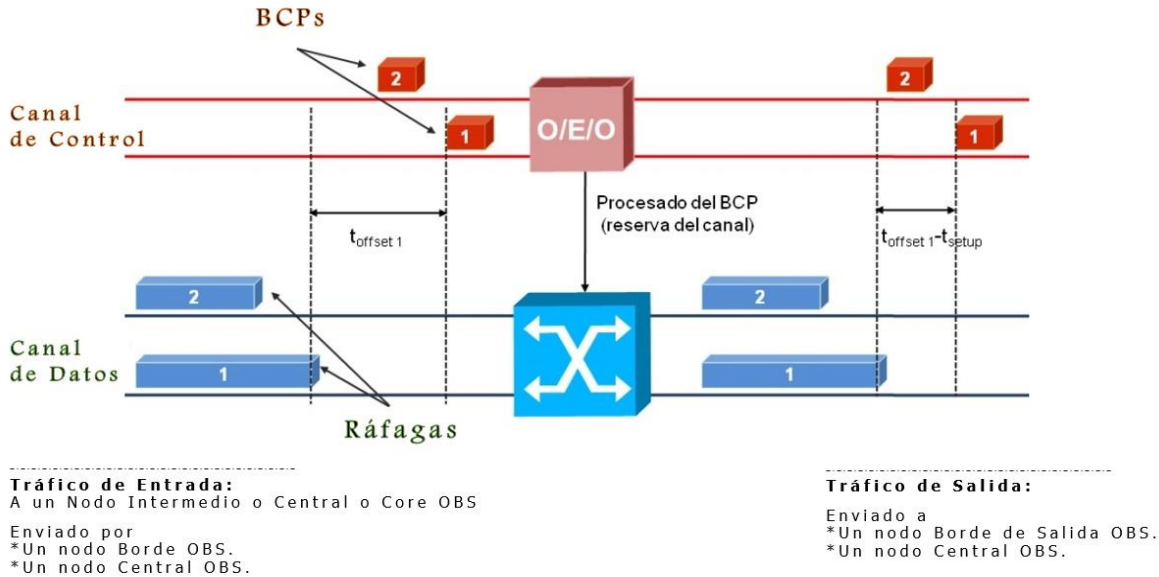


Figura 1. 22 Conmutación de Ráfagas en un Nodo Core

Fuente: Adaptada de [27]

1.3. 2 Proceso de Ensamble de Ráfagas -Módulos de Ensamblaje

El proceso de ensamble de ráfagas consiste básicamente en tres módulos, módulo clasificador, módulo ensamblador y el módulo planificador, ver figura 1.23 y figura 1.24, [9] [19] [20] [21] [27] [29] [30] [31]. En la tabla 1.4, se clasifica los diferentes módulos para ensamblar la ráfaga, según el objetivo, los procesos y la entrega o salida.

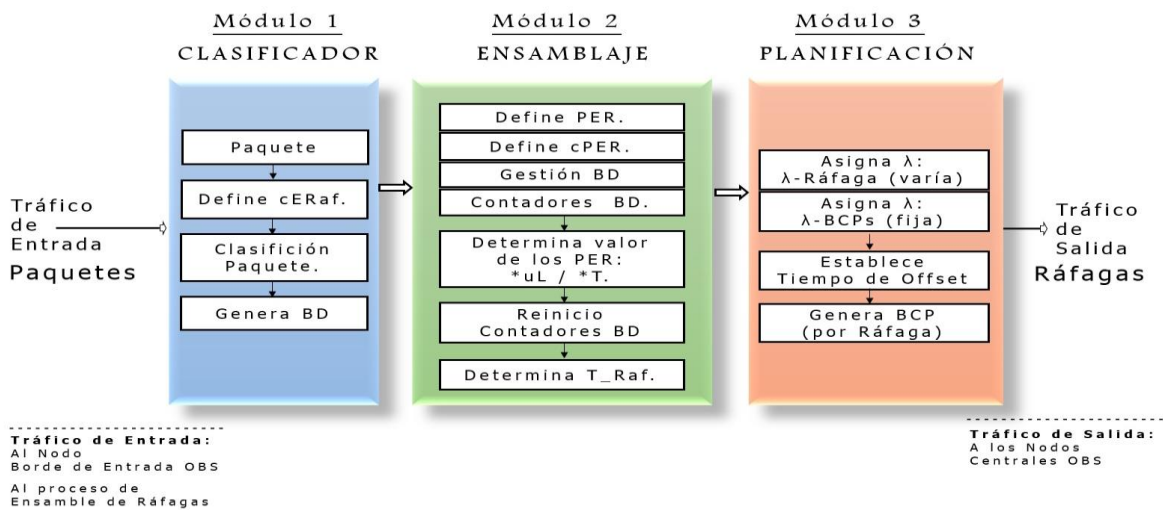


Figura 1. 233 Módulos del esquema Ensamble de Ráfagas

- Módulo Clasificador

El módulo clasificador, agrupa en colas o Buffers los paquetes que llegan al nodo borde de entrada según el criterio de ensamble de ráfagas (cERaf) seleccionado, el criterio base utilizado es la dirección de destino. Es por eso, que estas colas son llamadas Buffer's Destino (BD, *Buffer Destination*). Otros criterios de ensamble utilizados son: diferenciación de servicios (DiffServ, *Differentiated Services*), calidad de servicio (QoS, *Quality of Service*), por dirección origen, o la combinación entre ellos.

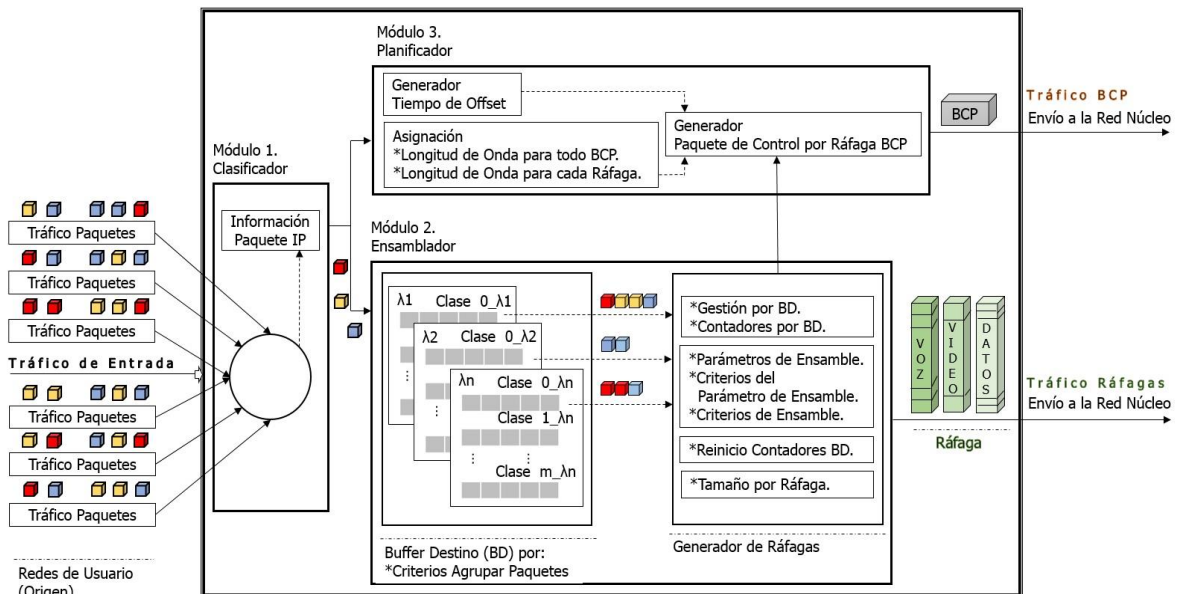


Figura 1. 244 Módulos del proceso de Ensamble en un Nodo Borde de Entrada OBS

Fuente: Adaptada de [9] [29] [30] [31]

- Módulo Ensamblador

La función principal del módulo ensamblador, es determinar el valor óptimo del parámetro de ensamble pre-establecido, ya sea el umbral de longitud (uL) o el tiempo de ensamblaje (T) o ambos, para asignar el tamaño adecuado a la ráfaga (macro-paquete) a transmitir. Teniendo en cuenta, cuando se implementa los dos parámetros de ensamble se utiliza el que primero ocurra. Para asignar el tamaño a la ráfaga se utiliza un contador de paquetes en cada uno de los BDs y cuando uno de los contadores alcanza el valor de umbral del parámetro, este umbral es determinado y establecido por el algoritmo de ensamble implementado, asigna dicho valor como tamaño de la ráfaga a transmitir y en ese instante todos los contadores son reinicializados.



Tabla 1. 4 Clasificación de los Módulos del proceso de Ensamble de Ráfagas.

PROCESO DE ENSAMBLE DE RÁFAGAS			
	Módulo 1 Clasificador	Módulo 2 Ensamblaje	Módulo 3 Planificador
Objetivos	Clasifica Paquetes de entrada al nodo OBS.	1. Determina y asigna: Valores a los Parámetros de Ensamble. 2. Genera la Ráfaga: *Administrando los BDs.	Planifica el envío de la Ráfaga Ensamblada.
Procesos	1. Analiza Paquetes -Información de destino: *DD (Dirección de Destino). 2. Define Criterios de Ensamble de Ráfagas (cERaf.): *DD. *DiffServ. (Diferenciación de Servicios). *QoS (Calidad de Servicio). 3. Clasifica cada Paquete según el cERaf. 4. Genera BD (Buffers Destino), según la Clasificación de cada Paquete.	1. Define Parámetros de Ensamble de Ráfagas (PER): *Umbral de Longitud (uL). *Tiempo de Ensamble (T). 2. Define Criterios de los PER (cPER): i. Adaptativo /NO-Adaptativo. *tráfico Fijo / Variable. ii. Predicción de Tráfico, por: *Nivel de Congestión. *Nivel de Pérdidas. 3. Administra los BDs /Genera la Ráfaga. *Usa los PER y los cPER. *Implementa el Algoritmo propuesto. *Tamaño de la Ráfaga (T_Raf.)	1. Asigna λ (λ : Canal en la Fibra Óptica) i. λ -BCPs fija. ii. λ -Ráfaga varía. Usa: Algoritmo Clásico de Asignación de la λ menos usada. 2. Establece Tiempo de Offset: *N-saltos (entre los nodos borde Entrada-Destino) *Tiempo Separación entre Ráfaga y BCP. *Tiempo Procesamiento y Reserva de Recursos. 3. Genera el BCP por Ráfaga, con: *DD. *T_Raf. * λ -Ráfaga. * λ -BCPs. 4. Transmisión del BCP.
Entrega	1. Al módulo de Ensamblaje: *Arreglo de Colas por Destino o BD. 2. Al módulo Planificador: *DD.	1. Al Conmutador: la Ráfaga generada. 2. Al módulo Planificador: el T_Raf. (Tamaño de la Ráfaga).	1. Al Transmisor: el λ -BCPs. 2. Al Transmisor: el BCPs. 3. Al Conmutador: la λ -Ráfagas.

- Módulo Planificador

El módulo planificador, es el encargado de gestionar el envío de cada una de las ráfagas, el cual genera el BCP utilizando la información determinada en los otros módulos, por ejemplo: en el módulo 1, la dirección destino de cada ráfaga a transmitir clasificadas en de cada BD y en el módulo 2, la longitud de la ráfaga a transmitir. Además, establece el tiempo de offset para cada par ráfaga-BCP, asigna a cada ráfaga su longitud de onda y a todos los BCP's asigna una misma longitud de onda preestablecida. Donde, el BCP es enviado en un periodo de tiempo antes de su ráfaga, para hacer la reserva necesaria de los recursos de red, el ancho de banda y la longitud de onda, en cada uno de los nodos centrales, mientras la ráfaga debe esperar en la cola de salida del nodo borde de entrada. Después de este periodo de tiempo de espera, también conocido como tiempo de Offset, la



ráfaga es transmitida del nodo borde de entrada hacia los nodos centrales hasta llegar al nodo borde destino correspondiente.

1. 4 ALGORITMOS DE ENSAMBLE DE RÁFAGAS

Todo algoritmo de ensamble de ráfagas propuesto en la literatura científica OBS se puede clasificar de forma general en algoritmos clásicos y no-clásicos. Los algoritmos de ensamble clásicos, se utilizan como mecanismo de contraste cuando se proponen nuevos algoritmos y se caracterizan principalmente porque implementan uno o los dos parámetros de ensamble, se clasifican en: (1) simples, cuando se implementa solo uno de los parámetros de ensamble, utilizando el valor del parámetro como tamaño de la ráfaga a transmitir, en la literatura se definen los algoritmos de ensamble basados en el tiempo (T) y los algoritmos basados en el umbral de longitud (uL); (2) mixtos, cuando implementa los dos parámetros de ensamble, pero utiliza el valor del parámetro que primero ocurra como tamaño de la ráfaga a transmitir. La figura 1.25, muestra el esquema de ensamble de ráfagas según el parámetro de ensamble.

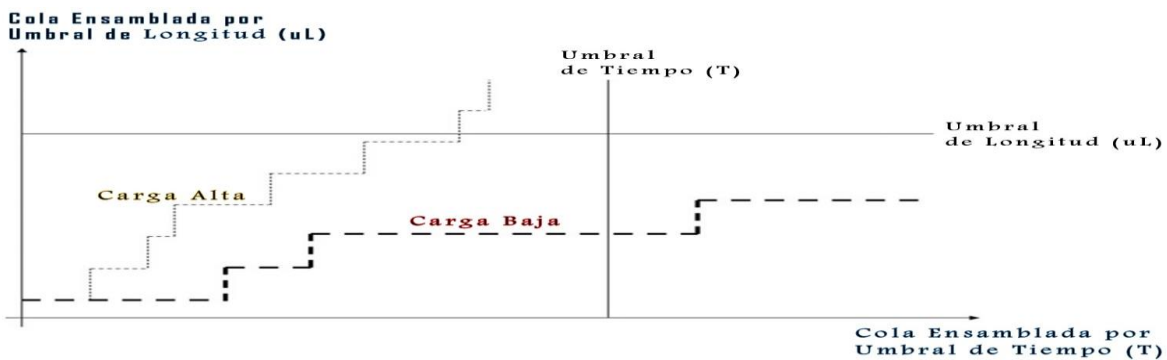


Figura 1. 25 Esquema de Ensamble de Ráfagas –Parámetros de Ensamble

Fuente: Adaptada de [27] [32]

Los algoritmos de ensamble no-clásicos o algoritmos de ensamble por combinación, tienen como objetivo principal determinar el esquema de ensamble que logre mejorar en cierto grado el desempeño total de una red OBS, utilizando la combinación de diferentes parámetros y criterios que inciden en el proceso de ensamble de ráfagas de datos y de esta manera mejorar una o más de las limitaciones de los algoritmos clásicos.

A continuación, se describe algunos de los algoritmos de ensamble por combinación existentes en la literatura OBS y se presenta una clasificación específica de 25 algoritmos de ensamble de ráfagas, clásicos y no-clásicos, teniendo en cuenta los principales parámetros y criterios que inciden en el proceso de ensamble, algunos de estos son: 1) parámetros de ensamble (PER); 2) criterios para agrupar los paquetes en los BD's o criterios al ensamblar la ráfaga (cERaf); 3) criterios de los parámetros de ensamble (Cper); 4) parámetros de desempeño,



otros. Esta clasificación se presenta como resultado del primer objetivo específico en este trabajo de grado de Maestría.

1.4. 1 Algoritmo de ensamble basado en el tiempo –T

El esquema de los algoritmos de ensamble basados en el umbral de tiempo o temporizador, considerado en la literatura científica como algoritmo de ensamble clásico o simple, establece un intervalo de tiempo límite (T) para ensamblar la ráfaga con una cantidad de n-paquetes y enviarla a su destino, ver figura 1.25. En [9] [26] [27] [30] [32] [33] [34] [35] [36] [37] [38] [39] [40] [42] [43] [44] [45] [46] [47] [48], cada uno de los autores, da un concepto general y/o proponen e implementan un nuevo algoritmo de ensamble basado en esta categoría. Sin embargo, establecer un valor óptimo de intervalo de tiempo de ensamble para tener un tamaño adecuado de ráfaga crea algunos inconvenientes. En particular, para el escenario con carga alta, presenta problemas en los nodos centrales debido a que incrementa la tasa de pérdida o probabilidad de bloqueo (PB) al generar ráfagas demasiado largas con tamaño variable. Por otro lado, el escenario con carga baja, genera ráfagas con tamaño medio-bajo donde el intervalo de tiempo llegará al umbral T antes de agregar suficientes paquetes en la ráfaga. Señalando, que la mayoría de estos esquemas utilizan como estrategia a la hora de ensamblar un proceso de relleno de bits para el envío de las ráfagas con un tamaño adecuado.

1.4. 2 Algoritmo de ensamble basado en el umbral de longitud –uL

El esquema del algoritmo de ensamble de ráfagas basado en el umbral de longitud, considerado en la literatura científica como algoritmo de ensamble clásico o simple, establece un valor de umbral uL , llamado en algunos casos tamaño mínimo (B_{min}), para agrupar un número mínimo de bytes en la ráfaga a ensamblar (en el BD correspondiente), ver figura 1.25. En [9] [26] [27] [30] [32] [33] [34] [35] [36] [37] [38] [40] [42] [43] [44] [45] [46] [47] [48], cada uno de los autores, da un concepto general y/o proponen e implementan un nuevo algoritmo de ensamble basado en esta categoría. Al ensamblar por uL , el escenario de carga baja presenta inconvenientes en el desempeño total de la red, debido a que incrementa el retardo de ensamblaje (D-ERaf) y el retardo extremo a extremo de transmisión de la ráfaga ($e2e-D$). Por otro lado, el escenario con carga alta genera ráfagas de tamaño fijo, el cual incrementa la probabilidad de bloqueo de las ráfagas ($PB_{ráfagas}$) en los nodos centrales de la red.

1.4. 3 Algoritmo de ensamble mixto

El algoritmo de ensamble de ráfagas mixto, determina el tamaño óptimo de la ráfaga a transmitir según el valor del parámetro de ensamble que primero ocurra en el proceso de almacenamiento en los BDs, ver figura 1.25. Esta categoría, el



algoritmo mixto, combina los beneficios de los dos esquemas de ensamble simples, del algoritmo de ensamble por umbral de T y del algoritmo de ensamble por uL. En [9] [26] [30] [32] [33] [34] [36] [37] [38] [40] [42] [43] [44] [45] [46] [47] [48], cada uno de los autores, da un concepto general y/o proponen e implementan un nuevo algoritmo de esta categoría. Sin embargo, el algoritmo mixto, también hereda sus problemas, por ejemplo, para carga de tráfico bajo (volumen del flujo de tráfico entrante bajo) genera un incremento en el retardo al ensamblar la ráfaga y en el retardo extremo a extremo, ya que los paquetes tienen que esperar hasta alcanzar el valor del umbral T o el valor del umbral uL. Esta categoría también es considerada en la literatura científica OBS como algoritmo de ensamble clásico o simple.

1.4. 4 Algoritmos de ensamble de ráfagas híbridos dinámicos basados en calidad de servicio y por servicios diferenciados

Los algoritmos de ensamble en esta categoría tienen como estrategia agrupar los paquetes entrantes a la red en cada uno de los BDs según su destino y además diferenciando los servicios de cada paquete para ensamblar la ráfaga a transmitir. En [30] [32] [33] [34] [37] [42], cada uno de los autores, da un concepto general y/o proponen e implementan un nuevo algoritmo de ensamble basado en esta categoría. Señalando que, la diferenciación de servicios es un requerimiento para asignar calidad de servicio (QoS, Quality of Service) a un tipo de tráfico al proporcionar un nivel de prioridad de transmisión. Algunos autores en esta categoría, en el proceso de ensamble han encapsulado al mismo tiempo en una ráfaga dos o más niveles de prioridad o han creado cada ráfaga con un nivel de prioridad diferente.

En algunos casos, el objetivo de la diferenciación de servicios consiste en mejorar la QoS de aquellas ráfagas con mayor prioridad, sacrificando en cierto grado las ráfagas de menor prioridad y de esta manera disminuir la probabilidad de pérdida de ráfagas (PB) en toda la red OBS.

1.4. 5 Algoritmos de ensamble de ráfagas híbridos dinámicos por predicción

En la literatura científica OBS los esquemas de ensamble, en algunos casos, son llamados híbridos porque implementan los dos parámetros de ensamble, pero, para este trabajo de grado el algoritmo propuesto es mixto ya que establece los dos parámetros de ensamble y es híbrido dado que implementa una técnica y una metaheurística para el proceso de ensamblaje, la lógica difusa y la PSO. Por otro lado, un algoritmo de ensamble es dinámico, dado que los parámetros de ensamble toman un valor diferente de acuerdo a la carga del tráfico entrante. Además, esta categoría utiliza un mecanismo de predicción de tráfico como estrategia para anticipar el valor del o los parámetros de ensamble establecidos y generar la ráfaga a transmitir con un tamaño estimado. En [32] [33] [34] [41] [42]



[43] [44], cada uno de los autores, da un concepto general y/o proponen e implementan un nuevo algoritmo de ensamble basado en esta categoría. Sin embargo, cuando el tamaño asignado a la ráfaga por predicción es menor al tamaño óptimo real, será deficiente el aprovechamiento del recurso de ancho de banda y es posible que se genere un incremento en la probabilidad de bloqueo.

En esta categoría los algoritmos de ensamble se diferencian dependiendo a cuál parámetro de ensamble se le aplica el método de predicción, al uL o a T o a los dos. Por ejemplo, cuando se implementa el mecanismo de predicción solo al umbral T, este proceso puede aumentar el tiempo de Offset y por ende incrementar la probabilidad de pérdida de ráfagas (PB_ráfagas) en la red.

1. 5 CLASIFICACIÓN DE LOS ALGORITMOS DE ENSAMBLE

En la tabla 1.6, se presenta la clasificación de algunos de los esquemas o algoritmos o mecanismos de ensamble de ráfagas existentes en la literatura científica OBS, con base a una revisión y selección de artículos en la etapa estado del arte del presente trabajo de grado de maestría. Donde, en la tabla 1.5 se presentan los acrónimos de la tabla 1.6. Señalando, que dicha clasificación es un poco más específica en comparación a los artículos encontrados y referenciados en este documento, dado que, se tiene en cuenta los principales parámetros y criterios que inciden en el proceso de ensamble de ráfagas.

Tabla 1. 5 Lista de acrónimos de la tabla 1.6

Columna	Acrónimo	
	Ref.	Número de las "Referencias" Bibliográficas.
Tipo de Tráfico	Var.	Variable.
Parámetro de Ensamble (PER)	uL T	Umbral de Longitud. Intervalo de Tiempo de Ensamblaje.
Criterios PER	N-Adp Adp	NO Adaptativo. Adaptativo.
Criterios de Ensamble de Ráfagas	Tam. DiffServ. Predicc. N.Congest QoS	Tamaño de la Ráfaga. Servicios Diferenciados. Predicción Nivel de Congestión. Calidad de Servicio.
Parámetros de Desempeño	e2e-D PB ERaf.-D	Retardo Extremo a Extremo. Probabilidad de Bloqueo. Retardo al Ensamblar la Ráfaga.
Otra Función OBS	RWA	Enrutamiento y Asignación de Longitud de Onda.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

Tabla 1. 6 Clasificación de algunos de los Algoritmos, Técnicas y Mecanismos basados en el Ensamble de Ráfagas

Año	Ref.	Tipo de Trafico		Parámetros de Ensamble (PER)			Criterios PER			Criterios de Ensamble de Ráfagas				Parámetros de Desempeño				Otra Función OBS				
		Fijo	Var.	uL	T	mixto	N-Adp	Adp	Tam.	DiffServ.	Predicc.	N.Congest.	QoS	e2e-D	PB	ERaf.-D	Otro	Offset	Contención	Enrutamiento	RWA	Señalización
2000	53		√		√			√	√							√	√					
2001	49		√	√				√		√		√		√								√
2002	54		√		√			√	√					√		√						
2002	48		√	√				√		√		√		√				√				
2002	51		√	√				√	√			√	√	√		√					√	
2003	55		√		√			√	√					√			√					
2003	46		√	√			√		√	√		√		√				√				
2004	47		√	√				√		√		√		√				√				
2006	56		√			√		√			√		√	√		√						
2007	44		√	√				√	√				√									
2008	26		√			√		√	√			√		√					√			
2009	37		√		√			√	√	√		√	√	√	√							
2012	43		√			√		√					√			√	√					
2013	41		√		√			√	√				√	√		√						
2013	35		√		√			√	√				√	√		√						
2014	30		√		√			√	√				√	√								
2014	31		√			√		√	√				√	√								
2014	57	√			√		√		√				√				√					
2014	34	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
2015	40		√	√				√	√				√	√								
2015	33	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
2015	9		√			√		√	√			√	√	√			√	√			√	
2016	58		√	√				√	√	√		√	√	√								
2016	38		√		√			√					√	√		√						
2016	29		√	√				√	√	√		√	√	√								



CAPÍTULO 2. CLASES DE PROBLEMAS SEGÚN LA TEORÍA DE LA COMPLEJIDAD, FUNDAMENTOS Y CLASES DE METAHEURÍSTICAS, INTRODUCCIÓN A LA METAHEURISTICA POR OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS Y ELEMENTOS, OPERACIONES Y PRINCIPIO BÁSICO DE LA TÉCNICA LÓGICA DIFUSA

2. 1 CLASES DE PROBLEMAS –TEORÍA DE LA COMPLEJIDAD

El nivel de complejidad de un problema está relacionado con la complejidad del algoritmo o método de optimización a utilizar. La complejidad de un método de optimización está dada por la cantidad de operaciones que se deben realizar y el tamaño de los datos de entrada. Donde, estos dos factores llevan a la complejidad computacional. Por lo tanto, evaluar la eficiencia de un algoritmo se relaciona con la complejidad del mismo.

Por otro lado, para evaluar el nivel de un problema se utiliza la teoría de la complejidad. Esta teoría, se encarga de estudiar los recursos que requiere el proceso para resolver un problema. Donde, si el proceso es difícil de realizar tiende a ser complejo y necesita más recursos. En general, los recursos utilizados en cualquier proceso son: (1) tiempo, se define por el número de pasos que necesita el algoritmo o método para resolver un problema; (2) espacio, se define como la cantidad de memoria utilizada para resolver un problema [64] [65] [66] [67] [68] [69].

La teoría de la complejidad se clasifica según el número de operaciones, en:

- **Clase P (Polinomial)**

Los problemas de **Clase P**, es el conjunto de problemas de decisión que pueden ser resueltos mediante un algoritmo que funciona en un periodo de tiempo polinómico, en proporción a los datos de entrada (una cadena de caracteres de datos con longitud n), por una maquina secuencial determinista [64] [65] [66] [67] [68] [69]. Un problema P, por lo general son problemas computacionales resolubles, aunque también existen los que no son tratables en términos prácticos, por ejemplo, problemas que requieren al menos $n^{1000000}$ de operaciones.

- **Clase NP (No-Determinista en Tiempo Polinomial –Nondeterministic Polynomial Time)**

Los problemas de **Clase NP**, es el conjunto de problemas de decisión que pueden ser resueltos en un periodo de tiempo polinómico por un algoritmo no determinista.

El algoritmo no determinista consiste en dos fases: (1) fase de adivinación (no determinista), genera una salida que puede ser una posible solución al problema; (2) fase de verificación (determinista), bajo un proceso determinista prueba si la posible solución se ajusta a los requerimientos de una solución correcta y se entrega en un tiempo polinomial. Técnicamente se dice que el problema NP es fácil si se puede resolver en tiempo polinómico y es difícil si se resuelve en tiempo exponencial [64] [65] [66] [67] [68] [69].

- Clase NP –Completo

Los problemas de Clase **NP –Completo**, son problemas muy difíciles de resolver, no pueden ser resueltos en tiempo polinomial, a diferencia de los problemas clase P y NP, con lo cual la complejidad computacional crece más rápido que el problema mismo, de forma exponencial. Para determinar una solución a un problema NP-Completo, se requiere transformar o reducir a un problema de clase NP [64] [65] [66] [69].

- Clase NP –Difícil (NP –Hard)

Los problemas de **Clase NP –Difícil**, son problemas con un nivel de complejidad mayor que los problemas de clase NP –Completo, dado que, no se puede garantizar encontrar la mejor solución posible en un tiempo razonable. Todo problema NP-completo es NP-Difícil. Esta clase de problemas se clasifican como problemas de optimización combinatoria [64] [65] [66] [67] [68].

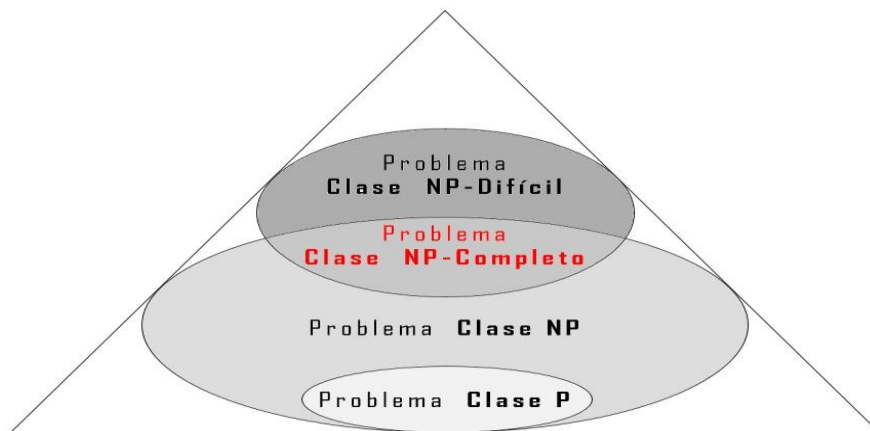


Figura 2. 1 Clases de Problemas según la Teoría de la Complejidad

Fuente: Adaptada de [69]

2. 2 METAHEURÍSTICAS -FUNDAMENTOS

Dado que, el proceso de ensamble de ráfagas, al igual que cada uno de los procesos del esquema funcional en las redes OBS, está catalogado como un problema NP –Completo o problema complejo, el cual, no permite encontrar un



tipo de solución trivial por medio de métodos o técnicas deterministas o resolverse en tiempo polinomial, donde, en la mayoría de los casos se necesita el uso de grandes ordenadores, con gran capacidad, por tal motivo se han desarrollado múltiples técnicas de optimización para tratar de resolverlos [70] [71] [72].

Los métodos de optimización se clasifican, ver figura 2.2, en: (1) exactos, garantizan encontrar una solución óptima para cualquier problema en un tiempo límite. El inconveniente está en el tiempo y memoria que necesitan, dado que, según el tamaño del problema estos crecen exponencialmente, en la mayoría de los casos esta técnica es inviable, ya que puede tomar mucho tiempo o una cantidad desorbitada de memoria; (2) aproximados, garantizan una solución adecuada o aproximada en un tiempo razonable, sacrificando la garantía de encontrar la solución óptima. [70] [72] [73].

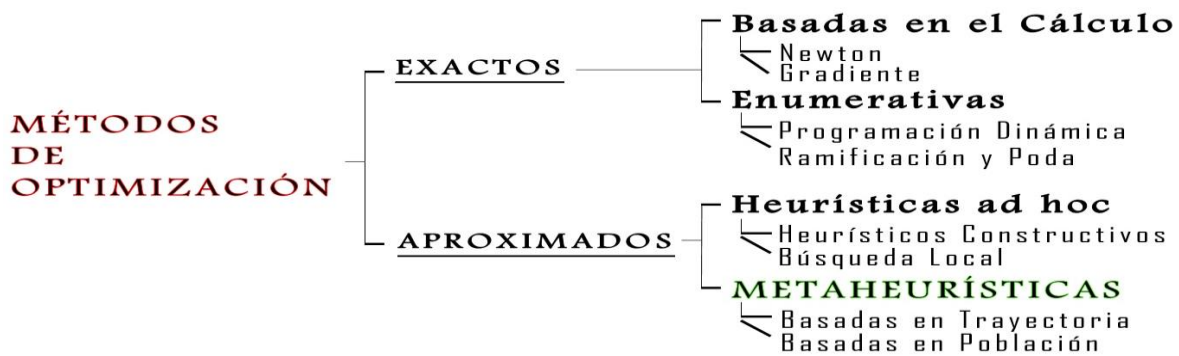


Figura 2. 2 Clasificación de los Métodos de Optimización

Fuente: Adaptada de [70]

Como se muestra en la figura 2.2, los métodos aproximados, son los heurísticos ad hoc y las metaheurísticas [70] [74] [75]. Los métodos heurísticos, se basan en un procedimiento simple, el cual, de un modo rápido, ofrecen una buena solución, aunque no necesariamente la óptima, para problemas complejos. Se dividen en heurísticos constructivos y de búsqueda local. Los métodos constructivos, determinan una solución completa partiendo desde cero, pero las soluciones ofrecidas suelen ser de muy baja calidad. Por otra parte, los métodos de búsqueda local parten de una solución ya completa y de esta manera realizan un proceso de búsqueda hasta encontrar un óptimo local, es decir, una solución mejor o igual que otra solución de su vecindario (valores próximos). El inconveniente se presenta en el proceso de búsqueda del vecindario, el cual se hace inabordable.

Las metaheurísticas, son algoritmos aproximados (no exactos), no deterministas y se basan en combinar diferentes métodos heurísticos con el objetivo de realizar un procedimiento simple de búsqueda de una forma más efectiva, para encontrar soluciones casi-óptimas, es decir, no garantizan la obtención del óptimo. Permiten trabajar con problemas de gran complejidad de una manera sencilla y obtener soluciones suficientemente buenas en tiempos razonables.

Características principales de las metaheurísticas:



- Son algoritmos aproximados, por lo tanto, no garantizan la obtención de la solución óptima.
- No saben si llegan a una posible solución óptima (o aproximada a la óptima). Por lo tanto, se les debe indicar cuándo deben detenerse.
- Son relativamente sencillos, se necesita de una solución inicial o conjunto de soluciones iniciales y de un mecanismo de revisión.
- El mecanismo de revisión puede aceptar malos movimientos. Algunas veces aceptan soluciones no factibles en un paso intermedio del proceso.
- La selección de la solución depende del instante del proceso, en general, la solución no tiene por qué ser necesariamente la misma.

La lógica de las metaheurísticas consiste: parte de una solución o conjunto de soluciones, que típicamente no es la óptima, se obtiene otras parecidas y se elige una que cumpla con algún criterio, a partir de la cual comienza de nuevo el proceso. Por lo tanto, este proceso se detiene cuando se cumple alguna condición establecida previamente.

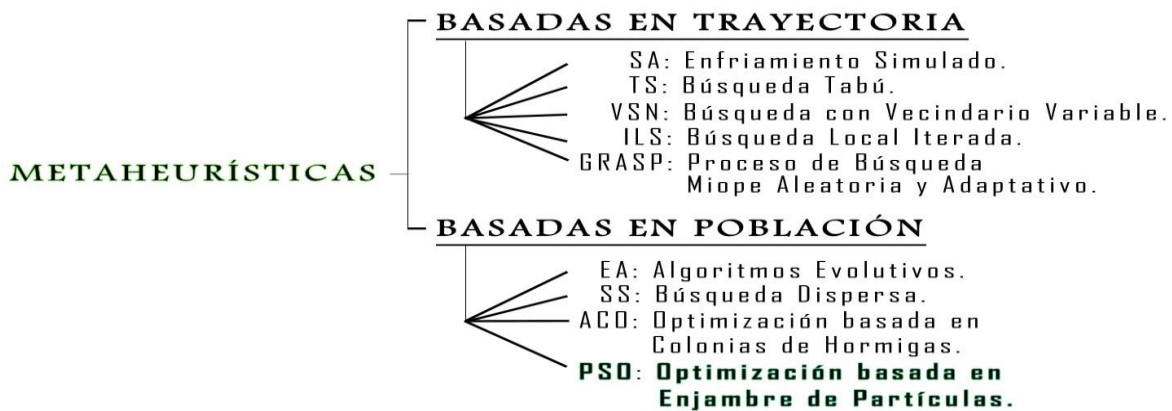


Figura 2. 3 Clasificación de las metaheurísticas

Fuente: Adaptada de [70]

Existen diferentes formas de clasificar las metaheurísticas dependiendo de las características que se seleccionen para diferenciarlas entre sí. En general, algunas clasificaciones recurren a cambios sucesivos de una solución a otra en la búsqueda del óptimo, mientras otras se sirven de los movimientos aplicados a toda una población de soluciones. Entre las taxonomías más representativas en la literatura de las metaheurísticas están: las inspiradas en la naturaleza y las no inspiradas, las basadas en trayectoria y las basadas en población, las basadas en memoria o sin memoria, con una o varias estructuras de vecindario, entre otras [70] [74] [75]. Este trabajo de grado, se centra puntualmente en uno de los métodos de optimización de las metaheurísticas basadas en población.



2.2. 1 Metaheurísticas basadas en trayectoria

Las metaheurísticas basadas en trayectoria, utilizan un único elemento del espacio de búsqueda, parten de una solución y mediante el proceso de exploración del vecindario (valores próximos), van remplazando la solución actual, formando una trayectoria, esta búsqueda termina cuando se alcanza un número predefinido de iteraciones y como resultado se puede encontrar una solución con una calidad aceptable [70] [74] [75]. Los métodos de optimización más representativos en las metaheurísticas basadas en trayectoria son:

- Enfriamiento simulado (SA).
- Búsqueda tabú (TS).
- Búsqueda con vecindario variable (VNS).
- Procesamiento de búsqueda miope aleatoria y adaptativo (GRASP).
- Búsqueda local iterada (ILS).

2.2.1 1 Enfriamiento simulado (SA)

El método de Enfriamiento Simulado SA (*Simulated Annealing*), aplica una estrategia para evitar obtener una solución que sea un mínimo local. Donde, en cada iteración, a partir de la solución actual R, se elige una solución R' del vecindario, donde R' tiene cierta probabilidad que sea peor que la solución actual. Entonces, si R es mejor que R', R' se sustituye a R como solución actual. Si R' es peor que R, se acepta, pero con una determinada probabilidad.

2.2.1 2 Búsqueda tabú (TS)

El método de Búsqueda Tabú TS (*Tabu Search*), es una de las metaheurísticas que se ha aplicado con mayor éxito para resolver problemas de optimización combinatoria. Las metaheurísticas TS, utilizan un historial de búsqueda para evitar los mínimos locales y para implementar su estrategia de exploración y de esta manera evitar buscar varias veces en la misma región. En cada búsqueda se elige la mejor solución y se añade a la lista tabú. La estrategia de exploración de la TS, se basa en crear una lista de los movimientos que ha generado cada solución, esta lista permite filtrar el vecindario, generando un conjunto reducido de soluciones.

2.2.1 3 Procedimiento de búsqueda miope aleatorizado y adaptativo (GRASP)

La metaheurística de búsqueda miope aleatoria y adaptativo GRASP (*Greedy Randomized Adaptive Search Procedure*), es un método de optimización simple que integra los métodos heurísticos constructivos con métodos de búsqueda local.



Los métodos a implementar puede ser simples o algoritmos más complejos como SA o TS. La GRASP consta de dos fases, primero la construcción de una solución y después un proceso de mejora. Esta mejora se obtiene debido al resultado del proceso de búsqueda. El heurístico constructivo aplica un mecanismo de construcción de soluciones y va añadiendo paso a paso diferentes componentes a la solución, de forma aleatoria.

2.2.1 4 Búsqueda local iterada (ILS)

La metaheurística de Búsqueda Local Iterada ILS (*Iterated Local Search*) es robusta, simple de implementar y altamente eficiente. La ILS, se basa en que cada iteración a la solución actual es perturbada para generar una nueva solución (solución intermedia). A esta solución, se le aplica una heurística que suele ser un método de búsqueda local para poder mejorarla. Este mínimo local obtenido puede ser aceptado como nueva solución actual si pasa por un test de aceptación. El método de perturbación debe generar una solución que no debe estar lejos del actual para que no sea una solución aleatoria, por lo tanto, debe utilizar un criterio de contra-balance, el cual filtra la aceptación de las nuevas soluciones dependiendo de la historia de búsqueda y sus características.

2.2. 2 Metaheurísticas basadas en población

Las metaheurísticas basadas en población, en cada iteración, trabajan sobre un conjunto de elementos del espacio o conjunto de soluciones, denominado población, a diferencia de las metaheurísticas basadas en trayectoria, trabajan con una solución del espacio de búsqueda [70] [74] [75]. Los métodos más representativos de las metaheurísticas basadas en población son:

- Búsqueda dispersa (SS)
- Algoritmos evolutivos (EA)
- Optimización basada en colonias de hormigas (ACO)
- Optimización basada en Enjambre de partículas (PSO)

2.2.2. 1 Búsqueda dispersa (SS)

La metaheurística por Búsqueda Dispersa (SS, *Scatter Search*) utiliza un conjunto pequeño de soluciones tentativas, llamado conjunto de referencia RefSet. Estos conjuntos se caracterizan por contener soluciones de calidad y estar distantes en el espacio de búsqueda. El proceso de la metaheurística SS tiene cinco pasos: (1) crear la población inicial; (2) generación del conjunto de referencia; (3) generación de subconjuntos de soluciones; (4) método de combinación de soluciones; (5) método de mejora.



El esquema general de la SS consiste: comienza creando un conjunto inicial de soluciones diversas en la fase de inicialización, donde, de forma iterativa genera nuevas soluciones, estas nuevas soluciones pasan al método de mejora, el cual, aplica el proceso de búsqueda local, y la solución resultante se inserta en la población inicial. Al final de la fase de inicialización, se aplica el bucle principal de la búsqueda dispersa. En este punto, se construye el conjunto de referencia a partir del conjunto inicial de soluciones usando el método de actualización del conjunto de referencia. Las soluciones del conjunto de referencia se agrupan en subconjuntos mediante el método de generación de subconjuntos. Donde, las soluciones de cada subconjunto son combinadas de alguna manera para producir nuevos individuos. La forma de combinar estas soluciones la define el método de combinación. Finalmente, el método de mejora se aplica a cada nueva solución generada y se decide cuáles de ellas se insertan en el conjunto de referencia.

2.2.2. 2 Algoritmos evolutivos (EA)

Las metaheurísticas por algoritmos evolutivos EA (*Evolutionary Algorithms*), están inspirados en la evolución natural, utilizan un proceso iterativo y estocástico que opera sobre una población de soluciones, denominadas individuos. El proceso completo del esquema de los EA es repetido hasta que se cumpla un cierto criterio de terminación, normalmente se utiliza el número dado de iteraciones. El esquema general de los EA, en la fase de selección, crea una población inicial de forma aleatoria, en algunos casos utilizando un heurístico de construcción, donde se escogen los individuos más aptos de la población actual para ser recombinados en la fase de reproducción. En la fase de reproducción, los individuos resultantes de la recombinación se alteran mediante un operador de mutación. Finalmente, en la fase de reemplazo, se genera la nueva población a partir de la población actual y/o los mejores individuos generados.

2.2.2. 3 Optimización basada en colonias de hormigas (ACO)

Las metaheurísticas por Optimización basada en Colonias de Hormigas ACO (*Ant Colony Optimization*), están inspirados en el comportamiento de las hormigas reales cuando buscan comida. El esquema de búsqueda de comida utilizado por las hormigas, está definido en: (1) Exploración, mecanismo de búsqueda en su proximidad, en el área; (2) camino por feromonas, la hormiga deposita una sustancia química denominada feromona, esto ayuda al resto de hormigas a encontrar la comida. Este método, camino por feromonas, capacita a la hormiga para determinar el camino más corto entre la comida y el nido. Las metaheurísticas ACO, para resolver problemas de optimización se basan en dos pasos, en la construcción de una solución basada en el comportamiento del esquema de búsqueda de comida utilizado por las hormigas y en la actualización de los rastros de las feromonas artificiales.



2.2.2. 4 Optimización basada en enjambres de partículas (PSO)

Las metaheurísticas por Optimización basada en Enjambres de Partículas (PSO, *Particle Swarm Optimization*), fue introducido por James Kennedy y Russell Eberhart en 1995, han sido usadas en diferentes campos, fundamentalmente, para la solución de problemas de optimización, recientemente se utiliza para problemas combinatorios. Están inspiradas en el comportamiento social de un conjunto o grupo de animales, donde cada elemento del conjunto es considerado partícula; por ejemplo, los más comunes: el vuelo de las bandadas de aves y el movimiento de los bancos de peces [70] [74] [75].

El esquema general de las metaheurísticas PSO, consiste en un enjambre o conjunto de partículas que son un conjunto de soluciones inicializadas aleatoriamente en el espacio de búsqueda, cada solución es considerada partícula, las cuales, para encontrar el óptimo actualiza las distintas generaciones que se tengan. Cada partícula, posee una posición y una velocidad que van cambiando constantemente conforme avanza la búsqueda, utilizando actualización de velocidad y actualización de la posición. La calidad de la posición de la partícula es evaluada por la función fitness predeterminada, esta puede ser la función objetivo o un criterio de optimización. Las partículas, van por el espacio de búsqueda con una velocidad (conjunto de partículas en movimiento), que es influenciada por la mejor posición encontrada por él hasta el momento y por la mejor posición encontrada por sus vecinos [70] [74] [75] [76] [77] [78] [79] [80].

Por lo tanto, el proceso de la PSO en cada iteración, gestiona, optimiza y entrega dos parámetros. Al alcanzar el número de iteraciones especificado u otro tipo de criterio de parada, el enjambre converge a la solución más apropiada, cercana a la óptima, para el problema.

Para el diseño del algoritmo de ensamble de ráfagas propuesto en este trabajo de grado de Maestría, se analizó cada una de las metaheurísticas existentes en la literatura y se encontró que la metaheurística de Optimización basada en Enjambre de Partículas (PSO) es la más adecuada para solucionar en cierto modo el problema seleccionado de estudio en una red OBS, el proceso de ensamble de ráfagas, ya que, se utilizan dos parámetros en el proceso y se entrega como resultado un valor próximo al óptimo de cada uno. Además, con base a la revisión bibliográfica del estado del arte de los diferentes métodos, algoritmos, mecanismos y/o técnicas, y combinaciones entre ellas, utilizadas para mejorar en cierto modo el proceso de ensamblaje, se elige, la técnica Lógica Difusa o técnica Fuzzy (FL, *Fuzzy Logic*), presentando una forma diferente de realizar el proceso de ensamble de ráfagas, algoritmo de ensamble propuesto, con el objetivo de mejorar en cierto grado el proceso en cuanto al desempeño de la red OBS. Señalando, que la PSO es la fase de optimización del proceso difuso en el algoritmo de ensamble propuesto. En la siguiente sección, se da los fundamentos y los principales elementos que se utiliza en la técnica Fuzzy (técnica FL).



2. 3 TÉCNICA LÓGICA DIFUSA

El concepto de lógica difusa fue planteado por Lofti Zadeh en 1965, quien al encontrar en los conjuntos clásicos el problema de utilizar solo dos opciones para procesar y clasificar información, la pertenencia o no de un elemento, la presentó como una forma de procesar información permitiendo valores parciales o borrosos a unos conjuntos, denominados conjuntos difusos. Donde, las operaciones que se pueden realizar con los conjuntos clásicos, corresponden a casos particulares de la matemática difusa (operaciones entre conjuntos difusos) y de la misma forma, la lógica clásica se puede considerar como un caso particular de la lógica difusa. Por ejemplo, la lógica difusa, en pocas ocasiones tiene en cuenta que el concepto de blanco/negro o verdadero/falso es absoluto, por el contrario, existen diferentes tonos de gris o valores de verdad [81] [82] [83] [84] [85] [86] [87] [88] [89].

La diferencia entre un conjunto clásico (convencional) y un conjunto difuso, está en que un conjunto clásico divide sus elementos en el grupo de inscritos y los no inscritos. Esta división tiene la característica de ser definida, no ambigua, y es válida para algunos conjuntos; por ejemplo, la pertenencia de una letra en el conjunto de las vocales. Sin embargo, la idea básica de un conjunto difuso, es que un elemento forma parte de un conjunto con un determinado grado de pertenencia; por ejemplo, un número determinado de personas, se puede clasificar por su edad o por su altura, u otra característica. Para el caso del número de personas clasificadas por la edad, se puede dividir en tres conjuntos: de corta edad, edad promedio y de gran edad. De manera que, se puede considerar fronteras entre cada conjunto, la cual, tienen una transición gradual desde la no pertenencia (no inscritos) hasta la pertenencia (inscritos) total.

La lógica difusa en la práctica es de gran valor principalmente en los siguientes casos:

- Situaciones complejas, donde tener un modelo matemático es prácticamente imposible. Esto puede ocurrir, por la complejidad del problema o a que no se tiene suficiente información.
- Situaciones complejas, donde es muy costoso obtener una alta precisión.
- Situaciones complejas, donde puede requerir demasiado tiempo obtener una alta precisión.
- Situaciones, donde al utilizar los métodos clásicos se obtiene un modelo que no representa una adecuada simplificación del problema.

Los elementos básicos de la técnica lógica difusa, son:

- Número Difuso, Intervalo Difuso y Conjunto Difuso.
- Variables Difusas.
- Variables Lingüísticas y Conceptos Lingüísticos.



- Función de Pertenencia.
- Operaciones entre Conjuntos Difusos

2.3 1 Número Difuso, Conjunto Difuso e Intervalo Difuso

Se define un número difuso como un conjunto normalizado, cuya función de pertenencia es al menos continua a trazos y tiene un valor funcional $A(x) = 1$, justo para un elemento, ver figura 2.4 (a). Por lo tanto, un conjunto difuso (intervalo difuso) se puede definir como el conjunto de números difusos, donde a cada número le corresponde un valor de pertenencia que es asignado por una función de pertenencia, ver figura 2.4 (b), [81] [82] [83] [84] [85] [87] [88] [90]. Para las operaciones aritméticas en números difusos se permite trabajar con las operaciones aritméticas ordinarias solo para intervalos cerrados. La figura 2.4, en (a) muestra la diferencia entre un número ordinario 2.0 y un número difuso “cerca de 2.0”, y en (b) muestra la diferencia entre un intervalo ordinario [1.45, 2.64] y un intervalo difuso “cerca del intervalo [1.45, 2.64]”.

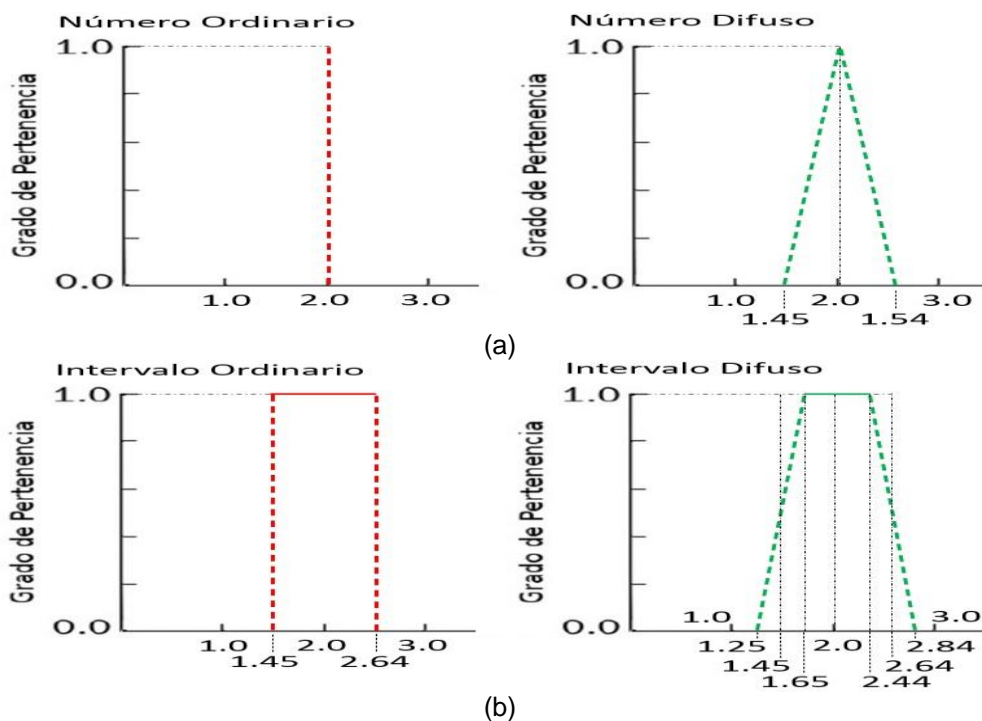


Figura 2. 4 Número difuso e Intervalo difuso

2.3 2 Función de Pertenencia

La función de pertenencia describe toda la información contenida en un conjunto difuso. Esta función asigna a cada elemento o variable difusa un valor real, que indica que tanto pertenece o en qué medida el elemento forma parte de ese conjunto difuso [81] [82] [85]. Un valor grande, denota un grado superior de

pertenencia a una categoría, mientras, valores pequeños denotan poca pertenencia. Esta función debe ser continua, cerrada y acotada. Las formas de las funciones de pertenencia más típicas son: la función triangular, la función trapezoidal y la función sigmoideal.

2.3.2. 1 Función de Pertenencia Triangular

Definida mediante el límite inferior **a**, el superior **b** y el valor modal **m**, tal que $a < m < b$, ver figura 2.5 y la ecuación 2.1. La función no tiene porqué ser simétrica.

$$\mu_A(x) = \begin{cases} 0, & \text{si } x \leq a \\ \frac{x-a}{m-a}, & \text{si } a < x \leq m \\ \frac{b-x}{b-m}, & \text{si } m < x < b \\ 0, & \text{si } x \geq b \end{cases} \quad (2.1)$$

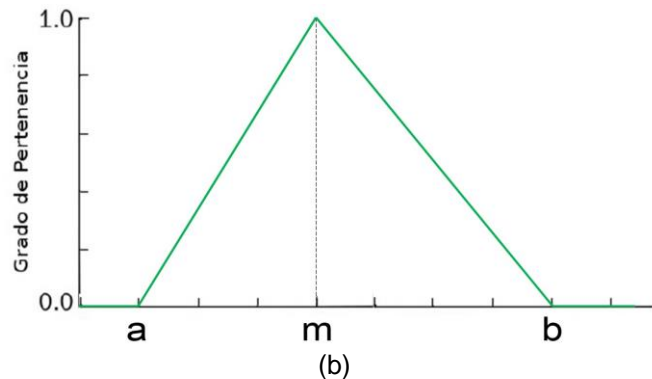


Figura 2. 5 Función de Pertenencia Triangular

2.3.2. 2 Función de Pertenencia Trapezoidal

Definida por su límite inferior **a**, límite superior **d**, y los límites de soporte **b** y superior **c**, tal que $a < b < c < d$, ver figura 2.6 y la ecuación 2.2. Si los valores de **b** y **c** son iguales, se obtiene una función triangular.

$$\mu_A(x) = \begin{cases} 0, & \text{si } (x < a) \text{ ó } (x > d) \\ \frac{x-a}{b-a}, & \text{si } a \leq x \leq b \\ 1, & \text{si } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{si } c \leq x \leq d \end{cases} \quad (2.2)$$

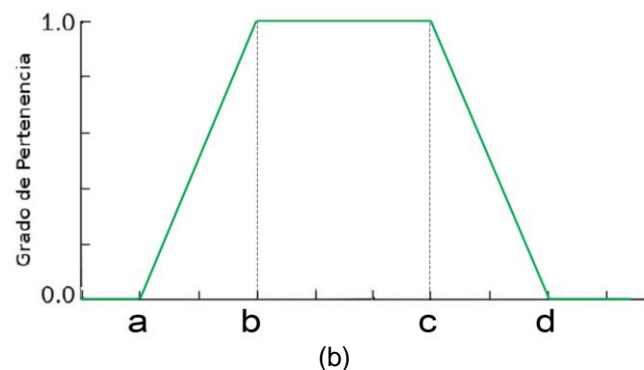


Figura 2. 6 Función de Pertenencia Trapezoidal

Casos especiales de la función trapezoidal son aquellas en las que algunos parámetros toman valores no finitos, ver figura 2.7, figura 2.8, figura 2.9 y la ecuación 2.3, 2.4 y 2.5 respectivamente, se tienen:



***Función Trapezoidal con parámetro $a = b = -\infty$**

$$\mu_A(x) = \begin{cases} 0, & \text{si } x > d \\ \frac{d-x}{d-c}, & \text{si } c \leq x \leq d \\ 1, & \text{si } x < c \end{cases} \quad (2.3)$$

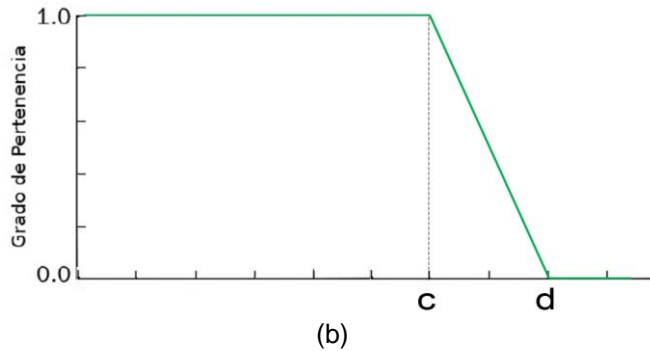


Figura 2. 7 Función de Pertenencia Trapezoidal $a = b = -\infty$

***Función Trapezoidal con parámetro $c = d = +\infty$**

$$\mu_A(x) = \begin{cases} 0, & \text{si } x < a \\ \frac{x-a}{b-a}, & \text{si } a \leq x \leq b \\ 1, & \text{si } x > b \end{cases} \quad (2.4)$$

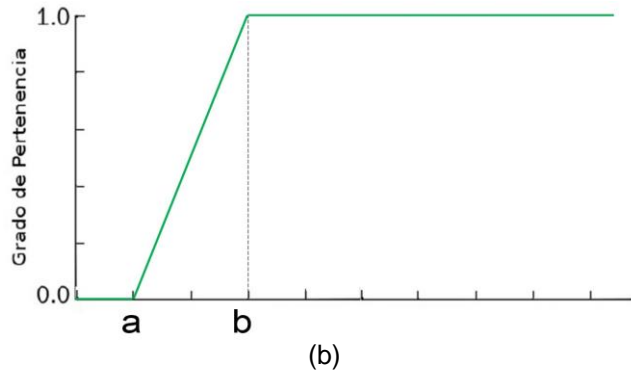


Figura 2. 8 Función de Pertenencia Trapezoidal $c = d = +\infty$

2.3.2. 3 Función de Pertenencia Sigmoidal

Definida por sus límites inferior a , superior b y el valor m o punto de inflexión, tales que $a < m < b$. El crecimiento es más lento cuanto mayor sea la distancia $a - b$. Para el caso concreto de $m = (a + b) / 2$ ver figura 2.9.

$$\mu_A(x) = \begin{cases} 0, & \text{si } x \leq a \\ 2 \left[\frac{x-a}{b-a} \right]^2, & \text{si } a < x \leq m \\ 1 - 2 \left[\frac{x-b}{b-a} \right]^2, & \text{si } m < x < b \\ 1, & \text{si } x \geq b \end{cases} \quad (2.5)$$

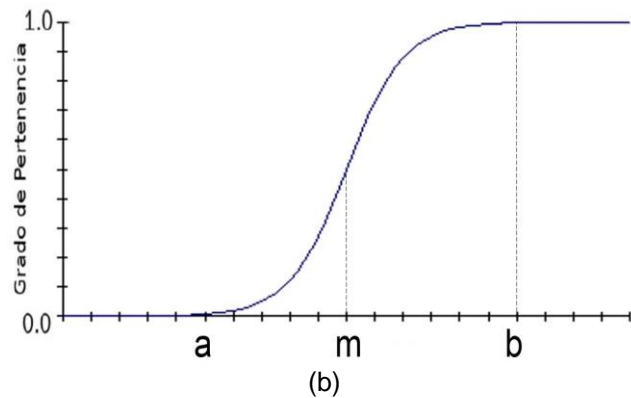


Figura 2. 9 Función de Pertenencia Sigmoidal

2.3 3 Variables Lingüísticas, Variable Difusa y Estado Difuso

La lógica difusa utiliza variables lingüísticas para definir los estados de una variable difusa [81] [82] [83] [85] [88]. Los estados difusos; por ejemplo, ver figura 2.10, un número de personas clasificadas por la edad, en el rango $[E1, E2]$, la edad se toma como la variable difusa, las variables lingüísticas son: (1) de Corta Edad A_1 ; (2) Edad promedio A_2 ; (3) de Gran Edad A_3 , donde, cada variable lingüística representa un estado de la variable difusa y cada estado difuso es un conjunto difuso. Además, cada estado difuso está definido por la función de pertenencia, que se puede expresar de la siguiente forma: $[E1, E2] \rightarrow [0,1]$.

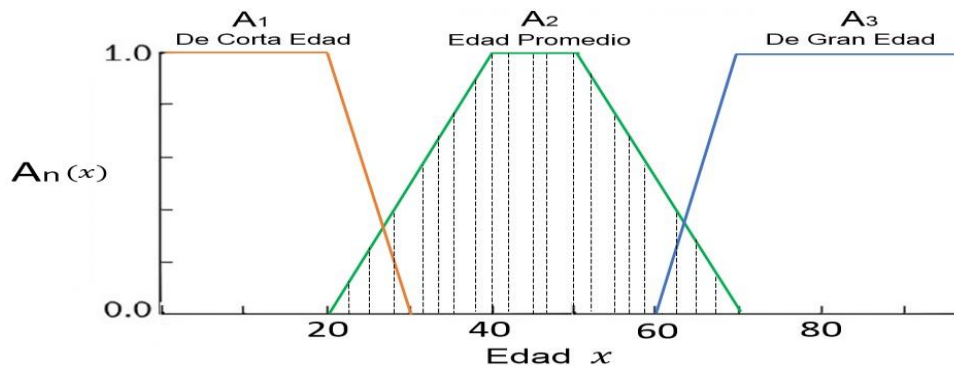


Figura 2. 10 Conceptos lingüísticos en cada una de las variables difusas y la función de pertenencia trapezoidal.

2.3 4 Operaciones entre Conjuntos Difusos

Las operaciones difusas consisten en la generalización de las operaciones básicas de los conjuntos clásicos, que son complemento, intersección y unión. [81] [82] [83] [84] [85] [89].

- **Unión:** Dado dos conjuntos difusos, A y B , la unión normal, $A \cup B$, ver figura 2.11 (a), se define para todo $x \in X$ por la ecuación:

$$(A \cup B)(x) = \max[A(x), B(x)] \quad (2.6)$$

- **Complemento:** el complemento de un conjunto difuso A con respecto al conjunto universo X , ver figura 2.11 (b), es definido para todo $x \in X$ por la ecuación:

$$\bar{A}(x) = 1 - A(x) \quad (2.7)$$

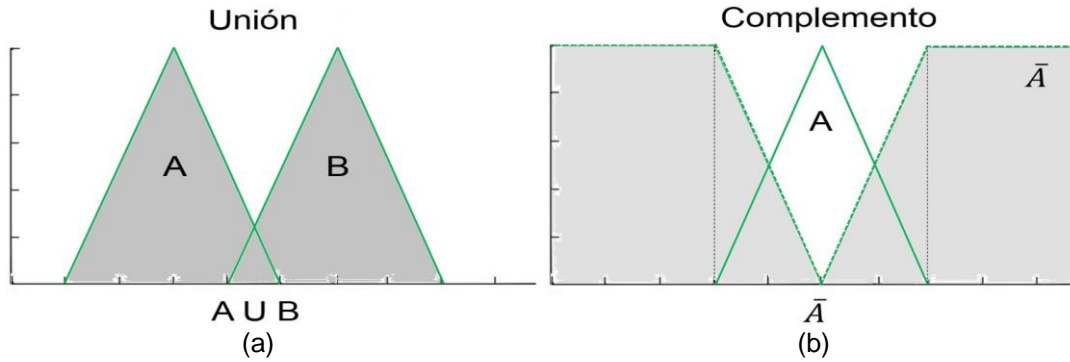


Figura 2. 11 Unión entre A y B y Complemento de A

Fuente: Adaptada de [81]

- **Intersección:** Dado dos conjuntos difusos, A y B , su intersección normal, $A \cap B$, ver figura 2.12, se define para todo $x \in X$ por la ecuación:

$$(A \cap B)(x) = \min[A(x), B(x)] \quad (2.6)$$

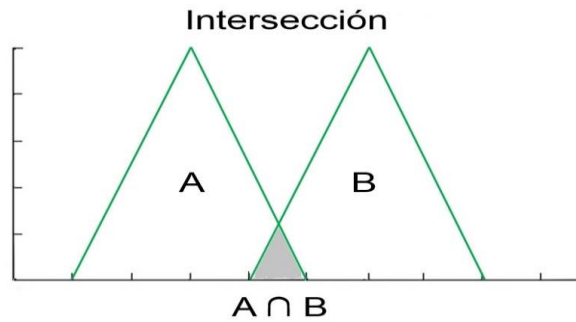
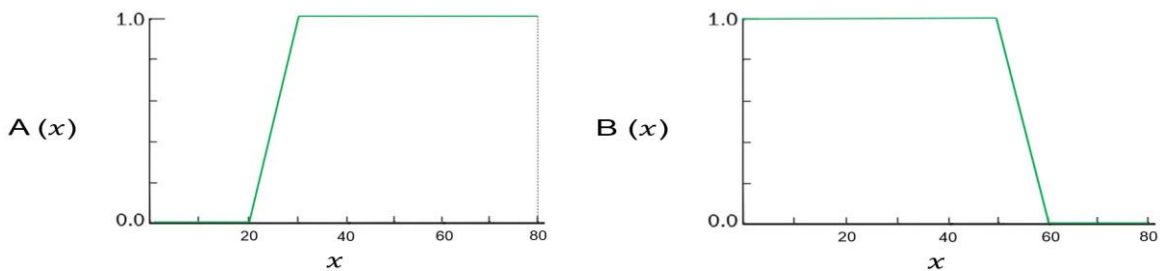


Figura 2. 12 Intersección entre A y B.

Fuente: Adaptada de [81]

La figura 2.13, muestra el ejemplo: número de personas clasificadas por edades, se utiliza la operación de intersección ($A \cap B$) para determinar e implementar la función de pertenencia y obtener el estado difuso de la variable lingüística edad promedio A2.



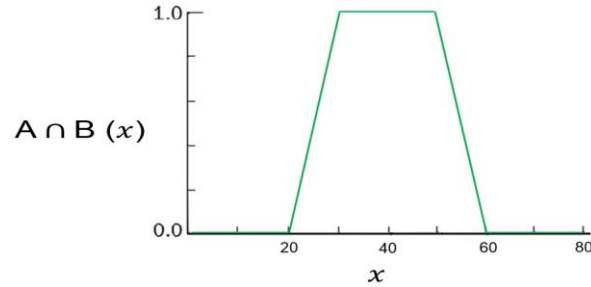


Figura 2. 13 Operación unión $(A \cap B)$, para determinar la función de Pertenencia del estado difuso A2

2.3 5 Principio básico de la técnica lógica difusa

En general el esquema funcional de la técnica lógica difusa o técnica fuzzy, mostrado en la figura 2.14, inicialmente asocia las variables de entrada con los valores de pertenencia en un conjunto difuso en el proceso llamado fuzzificación. La fuzzificación, es el proceso que transforma un conjunto clásico a un conjunto difuso, donde el valor discreto o valor Crisp (valor exacto y preciso) de la variable de entrada la transforma a un valor difuso. Una vez que se tienen los valores fuzzificados se trabaja con reglas lingüísticas, este proceso intermedio es llamado controlador difuso, el cual, utiliza leyes de control consistente en reglas lógicas, por ejemplo IF-THEN, formulando expresiones condicionales, en conjunto con funciones de pertenencia difusas, para al final obtener las variables de salida en el proceso de defuzzificación. La defuzzificación, es el proceso que transforma el valor de las variables de salida con valor difuso a su valor Crisp. [82] [83] [85] [87] [87] [89] [90].

La técnica fuzzy, proporciona una base matemática sólida para solucionar problemas complejos, modelando la incertidumbre de los procesos cognitivos humanos (razonamiento humano) al utilizar términos lingüísticos bajo ciertas reglas lingüísticas, de tal forma que permita ser entendido y tratado con precisión y en un tiempo adecuado por un ordenador o computador.

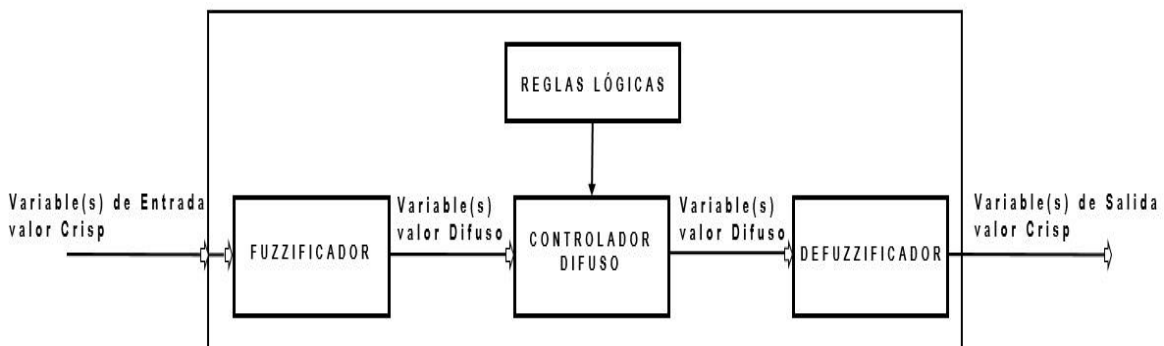


Figura 2. 14 Esquema funcional de la técnica Lógica Difusa

Fuente: Adaptada de [82] [83] [85] [86] [89] [90]



Estas reglas lingüísticas pueden ser muy descriptivas y fácilmente entendibles por un humano, pero difícilmente pueda ser representado en un idioma que sea capaz de interpretar un ordenador. Las palabras como “mucho”, “demasiado”, “muy poco”, utilizadas para interpretar cantidad, no tienen fronteras bien definidas, y en el momento de pasar a código se debe utilizar descripciones artificiales. Una forma de codificar es implementando conjuntos de intervalos; por ejemplo:

La cantidad de paquetes (volumen de tráfico) que llega a un nodo borde de entrada en una red OBS se interpreta utilizando los siguientes tres intervalos:

1. Intervalo volumen alto: de los 2'000.000 de paquetes a más.
2. Intervalo volumen medio: entre 800.000 a 2'000.000 de paquetes.
3. Intervalo volumen bajo: entre 1 a 800.000 paquetes.

Bajo la lógica convencional, representación Crisp, ver figura 2.15, para una carga de tráfico entrante de 1'999.999 paquetes, al emplear estos intervalos, el ordenador lo interpreta en el intervalo “volumen medio”, y si se incrementa en un paquete o más lo interpreta en el intervalo “volumen alto”. Por lo tanto, la lógica convencional determina un valor de pertenencia estricto en la frontera de cada conjunto o categoría.

Por otro lado, bajo la lógica difusa, ver figura 2.16, la cantidad de paquetes (volumen de tráfico) que llega a un nodo borde de entrada en una red OBS se puede interpretar utilizando:

- **Variable difusa:** cantidad de paquetes entrantes.
- **Variables lingüísticas:** tres variables lingüísticas = tres estados difusos. Son (1) volumen alto A1; (2) volumen medio A2; (3) el volumen bajo A3.
- **Conjuntos difusos:**
 - Intervalo volumen alto A1: de los 2'000.000 de paquetes a más.
 - Intervalo volumen medio A2: entre 800.000 a 2'000.000 de paquetes.
 - Intervalo volumen bajo A3: entre 1 a 800.000 paquetes.
- **Función de Transferencia:** función triangular para el estado difuso A2 y para los estados difusos A1 y A3 utiliza una función trapezoidal.
- **Grado de Pertenencia:** se determina utilizando la función de pertenencia en cada uno de los estados difusos. El caso crítico se presenta en las fronteras de cada uno de los estados difusos. En las fronteras y sus valores más próximos, se determina a que estado difuso pertenece una cantidad de paquetes seleccionando el de mayor grado de pertenencia.

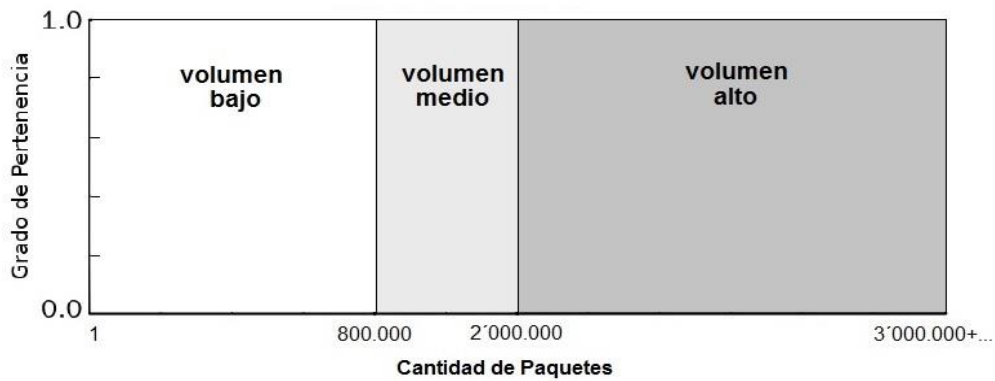


Figura 2. 15 Cantidad de paquetes entrantes a un nodo borde OBS vs el grado de pertenencia a un conjunto bajo la lógica convencional.

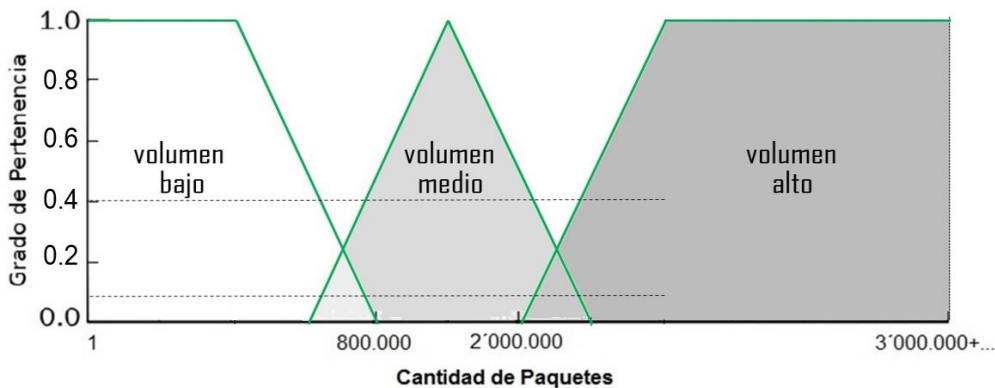


Figura 2. 16 Cantidad de paquetes entrantes a un nodo borde OBS vs el grado de pertenencia a un conjunto bajo la lógica difusa.

Partiendo, de que la lógica difusa en contraste a la lógica convencional, para determinar a qué estado difuso o conjunto difuso pertenece un elemento (partícula), proporciona una transición suave del grado de pertenencia entre los límites, casos críticos, ver figura 2.15 y figura 2.16, para el ejemplo de la cantidad de paquetes entrantes a un nodo borde OBS, saber a qué estado difuso pertenece el elemento 800.000 paquetes (cantidad de paquetes), este elemento está ubicado en la proximidad de una de las fronteras, se compara el grado de pertenencia de cada función y se selecciona el de mayor grado de pertenencia, por lo tanto, para 800.000 paquetes el ordenador interpreta que el tráfico está en el estado difuso volumen medio A2.

Por último, otras aplicaciones de sistemas basados en lógica difusa, diferentes a las redes de telecomunicaciones, son: (1) lavadoras, su sistemas de control determina automáticamente el ciclo de lavado según el nivel de suciedad y la cantidad de prendas [81] [89] [91]; (2) cámara digital, los estabilizadores de imágenes incorporan reglas que eliminan las vibraciones involuntarias de la persona [81] [89]; (3) control de temperatura, en [91] se presenta un ejemplo de control de temperatura para un invernadero, el cual, debe girar la válvula que



permite el paso de aire caliente para mantener una temperatura establecida, utilizando un control difuso. En [92] el autor propone un control de temperatura para un horno utilizando un control difuso; (4) metro de Sendai (Japón), tiene dos módulos, uno para el control de velocidad y otro para la parada automática [89]; (5) automóviles, transmisión automática difusa que ahorra combustible de un 12% a 17%, patentada por Nissan, además, desarrollo un sistema de frenos antilock (ABS) difuso [89].



CAPÍTULO 3. METODOLOGÍA DE DESARROLLO Y DE SIMULACIÓN, SELECCIÓN DE LA HERRAMIENTA DE SIMULACIÓN, TOPOLOGIA A IMPLEMENTAR EN LA RED OBS DE PRUEBA Y SUS MÓDULOS, DIAGRAMA DE FLUJO DEL ALGORITMO DE ENSAMBLE MIXTO Y DEL DMBA-FLPSO Y PSEUDOCÓDIGO DEL DMBA-FLPSO

3.1 METODOLOGÍA

3.1.1 Metodología de Desarrollo

Este capítulo contiene la descripción y caracterización del modelo de simulación de la red NSFNet OBS, abordando el problema de Ensamble de Ráfagas, para lo cual se describe la metodología empleada en el desarrollo de este trabajo de grado de Maestría. Esta metodología debe cumplir con las etapas de planificación, diseño y simulación del sistema, por lo que se propone la mostrada en la figura 3.1.

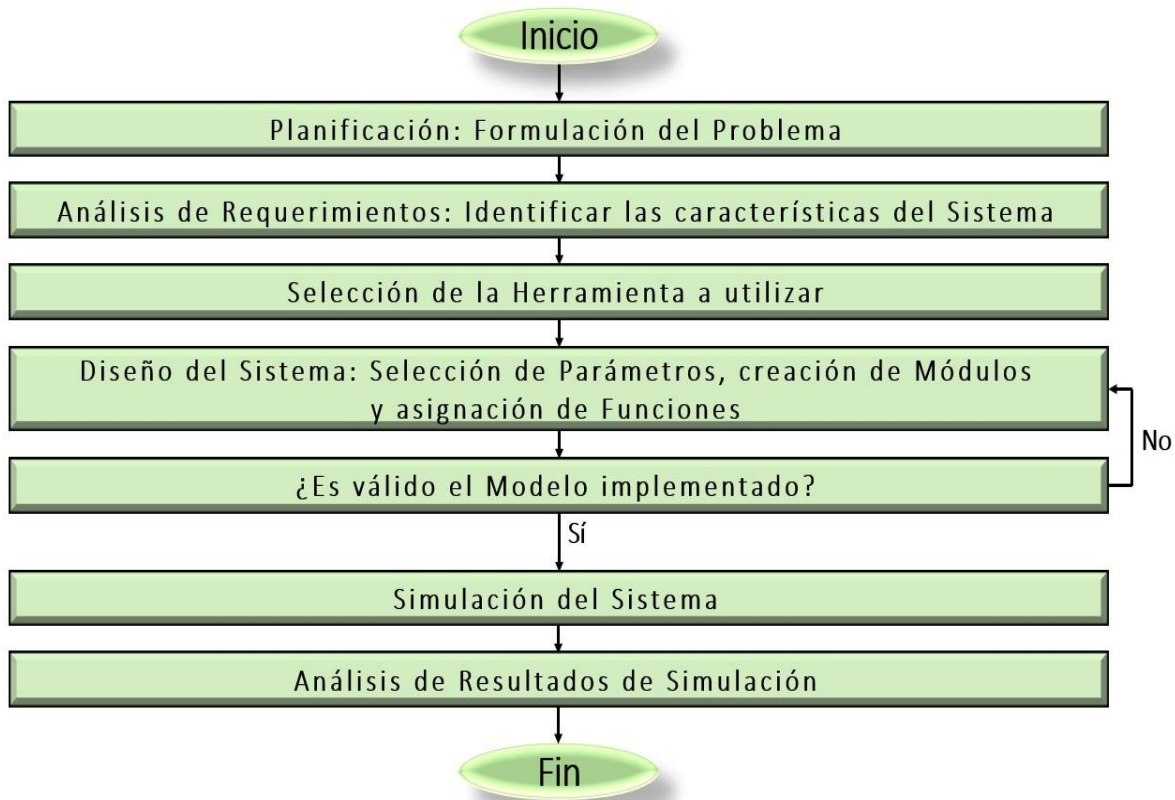


Figura 3. 1 Metodología aplicada para el desarrollo del trabajo de grado de Maestría.

Fuente: Tomada de [93]



La etapa de prueba o simulación del algoritmo desarrollado (propuesto), algoritmo de ensamble de ráfagas mixto dinámico basado en Lógica Difusa (FL) y por Optimización de Enjambre de Partículas (PSO) (**DMBAA-FLPSO**, *Dynamic Mixed Burst Assembly Algorithms–Fuzzy Logic and PSO*), comienza con la etapa de planificación, en la cual se formula el problema y al mismo tiempo se debe hacer un análisis de requerimientos para identificar las características del sistema. Una vez se cumplen estas etapas, se debe identificar las posibles herramientas a utilizar para la implementación de los módulos en la red. Posteriormente, se debe diseñar el modelo, implementarlo y verificar su correcto funcionamiento. Finalmente, se define los casos de estudio con el fin de ejecutar la simulación y analizar sus resultados.

3.1.2 Metodología de Simulación

Para definir la metodología de simulación, se tienen en cuenta los documentos “introducción a los simuladores” [94] del manual de OMNET++ y el documento “ingeniería del software, un enfoque práctico” [95]. En donde simulación se define como “*una técnica que imita el comportamiento de un sistema del mundo real conforme evoluciona en el tiempo*”.

Existen varias características mediante las cuales pueden ser clasificados los tipos de simulación, como son: la simulación estática, la cual representa un sistema en un solo instante de tiempo; la simulación dinámica, la cual representa la evolución de un sistema en diferentes instantes de tiempo; la simulación determinística, que no contiene ninguna variable aleatoria; la simulación aleatoria, la cual representa sistemas que contiene variables aleatorias; la simulación continua, donde el comportamiento del sistema cambia de forma continua en el tiempo; y la simulación discreta, donde el comportamiento del sistema cambia solo para instantes de tiempo concretos llamados eventos [94]. Dentro de la metodología de simulación, en [95] se definen un modelo adaptado para el desarrollo de las simulaciones, donde se detalla la metodología y sus etapas en términos de:

- **Definición del sistema:** realizar un análisis preliminar para determinar la posible interacción con otros sistemas, restricciones y variables involucradas.
- **Formulación del modelo:** definir los resultados que se esperan, precisar y construir el modelo que llevará a la obtención de los resultados.
- **Colección de datos:** definir con precisión y claridad los datos que requiere el modelo para llegar a los resultados deseados.



- **Implementación del Modelo:** definir el lenguaje de programación y/o herramienta de simulación a utilizar de acuerdo a las necesidades y objetivos del proyecto.
- **Validación:** determinar falencias en la formulación del modelo o las variables involucradas.
- **Interpretación:** análisis de resultados con el fin de implementar mejoras para generar mejores resultados.
- **Documentación:** de tipo técnica, en la que se describe el procesamiento de los datos y de tipo manual, en la que se describe el modelo desarrollado.

Por lo tanto, en la figura 3.2, para el desarrollo de la fase de simulaciones en este trabajo de grado de maestría se ha optado por aplicar una metodología con enfoque práctico, la metodología basada en el Proceso Unificado Racional –RUP (Rational Unified Process).

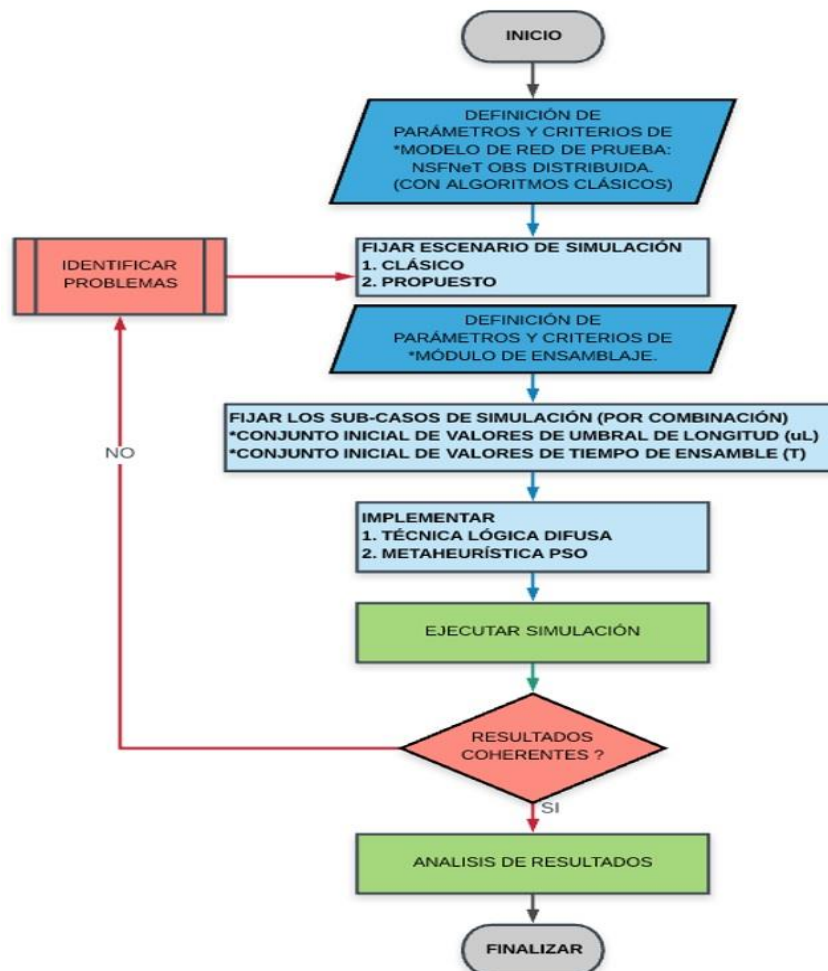


Figura 3. 2 Metodología aplicada para la Simulación de los Escenarios (casos) y SubCasos implementando el Algoritmo de Ensamble Clásico y Propuesto.



3. 2 SELECCIÓN DE LA HERRAMIENTA DE SIMULACIÓN

Para el desarrollo de este trabajo de grado de Maestría, es necesario utilizar una herramienta software que permita simular el comportamiento de una red OBS, implementar un algoritmo de ensamble dinámico basado en una metaheurística híbrida, integrando la técnica Lógica Difusa y la Metaheurística PSO sobre la topología NSFNeT, y de esa manera analizar el desempeño de la red en términos de la probabilidad de pérdida o probabilidad de bloqueo (PB) y el retardo extremo a extremo (e2e-D).

En esta sección se analizan las herramientas más reconocidas en el área de redes de telecomunicaciones que ofrecen el soporte y la precisión en cuanto a rendimiento y escalabilidad necesarias para el desarrollo de este trabajo de grado. A continuación, se presentan algunas de las herramientas que según la literatura han sido usadas para simular este tipo de redes:

3.2. 1 NS-2 (*Network Simulator 2*)

Es una de las herramientas de código abierto más confiable y de mayor uso para la implementación de proyectos en simulación. NS-2 se encuentra basado en eventos discretos, cuyas simulaciones están compuestas por código C++, usando lenguaje de comandos orientado a objetos para la definición de escenarios de simulación. Este simulador permite la configuración de gran cantidad de parámetros tales como la topología de red, la pila de protocolos y parámetros específicos de cada protocolo, para evaluar el impacto de diferentes tipos de tráfico de red. Las principales desventajas de NS-2, radican en que esta herramienta no posee interfaz gráfica, lo que ocasiona que el modelado sea una tarea complicada, además, el uso de scripts en la programación implica un aprendizaje previo por parte del usuario, lo que se traduce en mayor consumo de tiempo [96] [97].

3.2. 2 NS-3 (*Network Simulator 3*)

Esta herramienta es un simulador de eventos discretos de red de código abierto, que tiene como principales objetivos lograr un mayor énfasis en los niveles 2 y 4 del modelo OSI y que su uso sea principalmente educativo; a diferencia de NS-2, NS-3 está escrito solo en C++ lo que lo hace más fácil de depurar, además, esta herramienta ya posee una interfaz gráfica, la cual soporta algunas formas de animación visual para todo o parte de la simulación. Respecto a las tecnologías, NS-3 permite simulaciones sobre IPv4 e IPv6, Wireless (e.g., WiFi, Wimax) y algoritmos de enrutamiento (e.g., OLSR y AODV) [97].



3.2. 3 OPNET

Es una herramienta de simulación orientada a las comunicaciones. Presenta una interfaz amigable para los usuarios, incluyendo librerías de modelos que permiten la familiarización entre el programador y la jerarquía interna del simulador. OPNET (desde el 2012 llamado Riverbed) trabaja internamente con programación de código C++, el desarrollo de los modelos se da a través de la conexión de nodos, el cual se compone por diferentes módulos y conexiones [98].

Gracias a sus numerosas ventajas y a su poderosa interfaz gráfica, su uso, por parte de grupos académicos es alto, puesto que la manera de simular es muy intuitiva; sin embargo, necesita de gran cantidad de conocimientos previos en redes y programación, además es de licencia comercial [97].

3.2. 4 NCTUNS (National Chiao Tung University Network Simulator)

Es una herramienta de código abierto y se considera como uno de los mejores simuladores en las comunidades networking debido a su alta fiabilidad; además, porque NCTUns provee una interfaz de usuario integrada y profesional, por lo cual es una herramienta catalogada con un alto nivel de interfaz gráfica [97].

En cuanto al manejo que requiere la herramienta de simulación, se debe precisar que es necesario un conocimiento previo de tecnologías de red, ya que las implementaciones a las que está enfocada se orientan a comunicaciones en movimiento (e.g., VANETs y MANETs) [97].

Además, NCTUNS tiene la posibilidad de simular Wimax, VANET, MANET, Internet, Wireless LANS, GPRS Networks, Optical Networks, Personal AP, Real TCP/IP (Transmission Control Protocol, Protocolo de Control de Transmisión), UDP/IP. También, tiene la característica especial de simular CBR TCP, aplicaciones reales (generando datos reales) [97]. Por último, NCTUNS tiene un limitante: solo se puede implementar en Linux y además las últimas versiones el software no son gratuitas.

3.2. 5 OMNeT++

OMNeT++, es un Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*) que permite la simulación de redes de cualquier tipo, basado en componentes modulares y con un ambiente de simulación de arquitectura abierta y con un fuerte soporte de interfaz gráfica de usuario (GUI, *Graphical User Interface*). Es un simulador de eventos discretos, su principal aplicación es la simulación de redes de comunicación, y debido a su arquitectura genérica y flexible, ha sido utilizada en redes basadas en colas de espera, arquitectura de



hardware, tecnologías móviles, inalámbricas, redes ATM, TCP/IP, redes ópticas, entre otras [97] [99].

Esta herramienta consta de módulos escritos en C++, que se comunican entre sí a través de mensajes, donde, una cantidad n de módulos simples pueden conformar un módulo compuesto y los niveles jerárquicos no tienen límites [97]. Los módulos se pueden conectar entre sí a través de puertos y se comunican con el paso de mensajes a lo largo de rutas preestablecidas por las conexiones o directamente a su destino. Las conexiones se pueden utilizar para modelar enlaces físicos con parámetros como la velocidad de datos, tiempo de propagación, tasa de error de bit y tasa de paquetes perdidos [100].

Con respecto al diseño y modelado de la topología de red, OMNeT utiliza el lenguaje denominado Editor de Redes (NED, *Network Editor*), lenguaje de alto nivel para la creación de los módulos. Con éste, se puede establecer la topología de la red, los canales y sus parámetros, mensajes, interconexión de los módulos, entre otros.

Por otro lado, la interfaz gráfica que ofrece la herramienta presenta características de depuración superiores al de otras herramientas; además, OMNeT es un simulador con visualización en línea, lo que permite que los usuarios puedan pausar la simulación e inspeccionar o cambiar los valores de los módulos que conforman una red, cambiar la apariencia de los nodos y datos que arroja como resultado [97].

3.2. 6 Comparación entre las diferentes herramientas de simulación

En la tabla 3.1 se presenta una comparación de las diferentes herramientas de simulación expuestas anteriormente.

Con el fin de simular el comportamiento de una red OBS implementando un algoritmo mixto híbrido y dinámico para el ensamble de ráfagas, la herramienta más adecuada, según los requerimientos presentados en la tabla 3.1 es OMNeT++, ya que esta herramienta se caracteriza por ser flexible, de licencia libre, que presenta una buena interfaz gráfica, pero sobre todo permite la implementación de heurísticas, algo que las demás herramientas no soportan.

Además de la herramienta de simulación, utilizada para la creación de los módulos de la red OBS, es necesario contar con otro tipo de herramienta que permitan realizar gráficas a partir de los datos o resultados obtenidos en la etapa de simulación y de esa manera analizar el comportamiento del algoritmo de ensamble mixto híbrido y dinámico propuesto implementado en la red. A continuación, se presentan algunas de las herramientas más utilizadas para la representación gráfica de los datos, estas son:



- **MATLAB**

Es una herramienta de simulación que se utiliza como programa de cálculo matemático muy flexible y potente, tiene la característica de representación de datos gráficamente. Esta herramienta brinda la posibilidad de integrar nuevas funciones gracias a la amplia variedad de módulos con los que cuenta para ejecutar simulaciones con alto grado de complejidad [101].

- **ORIGIN PRO**

Es un software científico utilizado ampliamente en el ámbito académico para realizar gráficas de alta calidad. Este software cuenta con características de diseño avanzado por medio de los cuales se pueden realizar gráficos profesionales como la posibilidad de comparar diferentes datos establecidos, figuras en tercera dimensión (3D), las cuales facilitan el análisis de la información [102].

Tabla 3. 1 Requerimientos analizados sobre las herramientas de simulación

Fuente: Tomada de [97]

	NS-2	NS-3	NCTUNS	OPNET	OMNET
Uso Investigativo	Alto	Medio	Alto	Alto	Alto
Tipo de Licencia	Libre	Libre	Comercial	Comercial	Libre
Plataforma que soporta	Windows, Unix, Mac	Windows, Unix, Mac	Linux	Windows, Unix	Windows, Unix
Interfaz gráfica	Bajo	Medio	Alto	Alto	Medio
Graficación de resultados	No tiene	Aceptable	Aceptable	Buena	Aceptable
Tecnologías de nivel 2 y nivel 3 que soporta	Alto	Medio	Alto	Alto	Alto
Tráfico que permite modelar	Alto	Medio	Alto	Alto	Medio
Implementación de Heurísticas	No permite	No permite	No permite	No permite	Permite

De acuerdo a lo anteriormente expuesto, se considera que la herramienta más adecuada para la realización de las gráficas es ORIGIN PRO, debido a que es una herramienta fácil de utilizar y además permite realizar gráficas de alta calidad.

3. 3 MÓDULOS DE UNA RED OBS

Para implementar una red que cumpla con las características de OBS, se parte de la creación de los módulos simples, con el fin de formar los nodos de borde (*Edge Node*) y los nodos centrales (*Core Node*).

3.3. 1 *Nodo Fuente / Destino*

El nodo fuente (*source*), es el encargado de generar los paquetes que se desean transmitir por la red. Además, se encarga de asignar aleatoriamente la dirección de destino y el tipo de información que va ser enviada, las cuales, para este caso son asignadas siguiendo una distribución de probabilidad uniforme. Este módulo cuenta con los siguientes parámetros: dirección de origen, dirección de destino, tamaño de los paquetes (en bytes) y el tiempo entre generación de paquetes (en segundos), cuyas variables son declaradas en el archivo `.ned` del módulo y asignadas en el archivo de configuración `omnet.ini` del proyecto en OMNeT++.

Por otro lado, el nodo destino (*Sink*) recibe los paquetes provenientes de la red óptica y es donde se realizan los procesos estadísticos como el cálculo de retardo extremo a extremo (e2e-D) y el número de paquetes perdidos (probabilidad de bloqueo –PB).

El módulo fuente se agrupa con el módulo destino en un módulo de mayor jerarquía llamado *SourceSink* como lo muestra la figura 3.3.

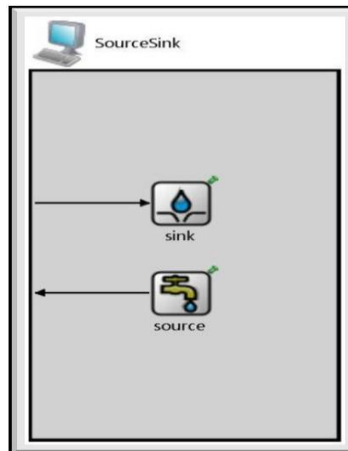


Figura 3. 3 *Modulo Fuente / Destino en OMNeT++. En un nodo borde OBS*

3.3. 2 *Nodo Borde (Edge Node)*

El nodo borde puede ser de Entrada (ingreso) o de Salida (egreso). En la figura 3.4, se muestra el esquema (en `.ned`) de los nodos de borde de entrada, dado por

el módulo de ensamble (Assembler) y los nodos borde de salida, dado por el módulo de desensamble (Disassembler), en OMNeT++, donde, el módulo del nodo de borde de entrada se agrupa con el módulo del nodo de borde de salida en un módulo de mayor jerarquía llamado EdgeNodeComplete.

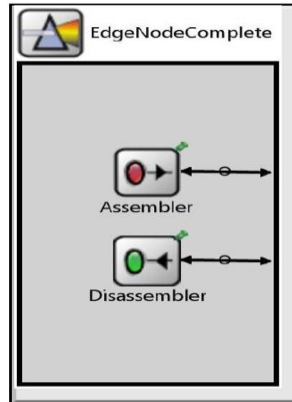


Figura 3. 4 Módulo del Nodo Borde de Entrada y de Salida

3.3.2. 1 Nodo Borde de Entrada (Assembler)

Como se dijo anteriormente, el nodo borde de entrada se encarga de recibir los paquetes provenientes de la fuente o redes externas (source), clasificarlos, ensamblarlos en la ráfaga y enviarla, para eso se ha creado el módulo llamado ensamblador (assembler). El cual, se compone de tres submódulos: un clasificador (classifier), un ensamblador de ráfagas (burstifier) y un planificador de envío (sender), como se muestra en la figura 3.5.

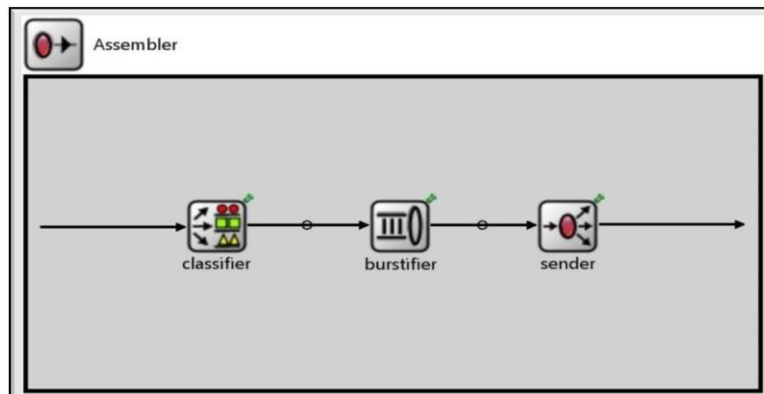


Figura 3. 5 Submódulos del Módulo de Ensamble en el Nodo Borde de Entrada

El clasificador (classifier) recibe los paquetes de datos provenientes de la fuente y los selecciona de acuerdo al criterio de clasificación, el cual, puede ser según su dirección de destino, el tipo de servicio, entre otros, para agruparlos en una cola o Buffer de datos. Una vez se haya clasificado los paquetes, estos se ensamblan en una ráfaga de datos en el submodulo conformador de ráfagas (burstifier).



El ensamblador de ráfagas forma la ráfaga con paquetes que tengan la misma dirección de destino (criterio de clasificación). Este proceso depende del algoritmo de ensamble implementado, que para este caso es el algoritmo de ensamble mixto adaptativo basado en la integración de la Lógica Difusa y la PSO, el cual, tiene en cuenta los dos parámetros de ensamble, el umbral de longitud (uL) y el tiempo de ensamblaje (T). Una vez la ráfaga alcanza el umbral de longitud o se cumpla el tiempo de ensamblaje, lo que primero ocurra, se detiene el proceso, se conforma la ráfaga y se envía hacia un submódulo llamado planificador de envío (sender), como se explicó en la sección del Proceso de Ensamble de Ráfagas –Módulos.

Una vez la ráfaga llega al planificador de envío (sender), este se encarga de generar el paquete de control de ráfaga (BCP), generar el tiempo de offset, realizar la reserva de recursos y planifica el envío del BCP y la ráfaga de datos hacia el nodo central. El paquete de control se llena con los siguientes parámetros: diferencia entre el tiempo actual y el tiempo estimado de llegada de la ráfaga, dirección de destino, color de la ráfaga o canal (longitud de onda) por la que es enviada la ráfaga, y el tamaño de la ráfaga.

Por otro lado, la reserva de recursos para la ráfaga se realiza mediante el algoritmo LAUC en la red OBS, donde, se busca el canal que se encuentre libre en el menor tiempo posible. Una vez que se tenga un canal de datos disponible para el envío de la ráfaga, se envía primero el BCP por un canal dedicado (en este caso corresponde al último) y después de un tiempo de offset calculado se envía la ráfaga.

3.3.2. 2 Nodo Borde de Salida (Disassembler)

Es el encargado de recibir las ráfagas de datos provenientes de otros nodos de la red, de los nodos núcleo (centrales), realiza el proceso de desensamblaje de la ráfaga y entrega los paquetes al destino (Sink), para lo cual, se ha creado el módulo Desensamblador (disassembler), ver figura 3.4. En este nodo se realizan los cálculos del número de ráfagas pérdidas o Probabilidad de Bloqueo (PB) y el retardo extremo a extremo (e2e-D) de ráfagas.

3.3. 3 Nodo Central (Core Node)

Los nodos de core o centrales (núcleo) permiten la conmutación de las ráfagas de datos de manera transparente sobre la fibra óptica, esto se hace configurando la matriz de conmutación para que cierta ráfaga que llega al conmutador salga por alguna de las interfaces de salida, dependiendo del destino que tenga y el canal de datos sobre el que viaja. Los nodos de núcleo están directamente conectados a los nodos de frontera mediante enlaces de fibra óptica, como se muestra en la figura 3.6.

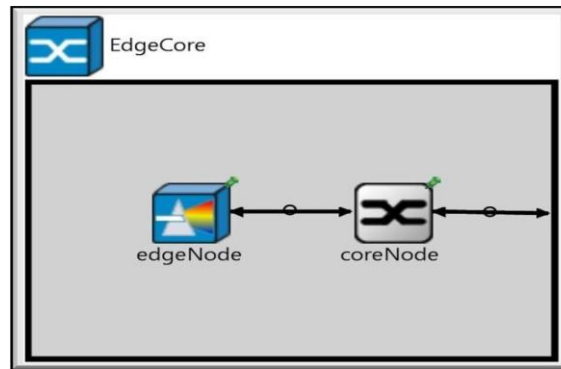


Figura 3. 6 Conexión entre el Nodo Borde y el Nodo Central en una red OBS

Un nodo central está compuesto por cuatro elementos: una interfaz de entrada (coreInput), un conmutador óptico (OXC), una unidad de control (ControlUnit) y una interfaz de salida (coreOutput) como se muestra en la figura 3.7. La interfaz de entrada se encarga de extraer el BCP y la ráfaga de datos; el BCP es enviado a la unidad de control, mientras que la ráfaga es enviada al conmutador óptico sin realizar ningún tipo de conversión. Por otro lado, la unidad de control recibe el BCP, realiza una conversión del dominio óptico a eléctrico (OE) y lo procesa con el fin de configurar el conmutador óptico para la llegada de la ráfaga, luego el BCP por conversión pasa al dominio óptico nuevamente (EO) y es enviado a la interfaz de salida, como se muestra en la figura 3.8. Así mismo, el conmutador óptico recibe la ráfaga de datos por un puerto de entrada específico y sale por un puerto que ya ha sido configurado por la unidad de control hacia la interfaz de salida. Por su parte la interfaz de salida envía el BCP por el canal dedicado y la ráfaga por su respectivo canal de datos hacia el siguiente nodo.

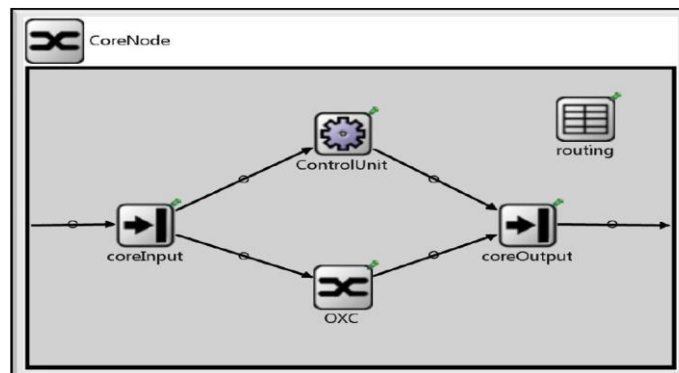


Figura 3. 7 Módulos en un Nodo Central

Además, de los módulos explicados anteriormente, el nodo central debe conocer información de enrutamiento para que el conmutador óptico envíe la ráfaga por el puerto adecuado, es por eso que se ha creado el módulo de enrutamiento (routing) basado en el algoritmo de dijkstra, como se mencionó en el capítulo 2. De igual manera, la unidad de control debe contar con la información de la asignación

de recursos, por lo que se crea el módulo GatesHorizon como se muestra en la figura 3.8.

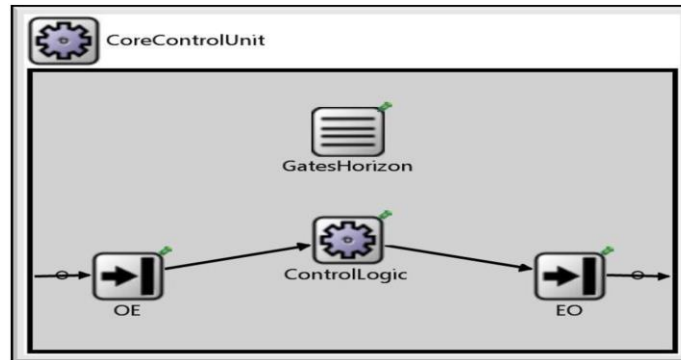


Figura 3. 8 Módulos de Unidad de Control en un Nodo Central

3.3. 4 Red OBS basada en la topología NSFNeT

Una vez creados cada uno de los módulos básicos de una red OBS como son los nodos borde y los nodos centrales, se procede a caracterizar la red, que para este caso tiene la topología de la NSFNeT (*National Science Foundation Network*).

NSFNeT es una red WAN que conecta 14 ciudades de Estados Unidos de América. La figura 3.9 muestra la topología de red con sus respectivas distancias físicas entre los nodos. Esta red es de gran importancia ya que la mayoría de pruebas se lleva a cabo sobre esta topología. Igualmente, es una de las topologías de red más empleadas para análisis de desempeño de redes en muchos trabajos de investigación. Esta red posee una topología distribuida irregular con su número de conexiones variables por nodo [103].

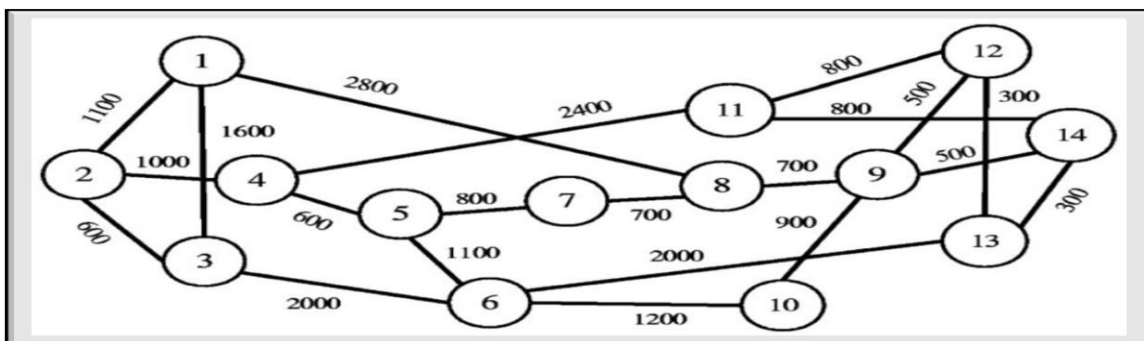


Figura 3. 9 Topología de la NSFNeT

Fuente: Tomada de [101]

La red explicada anteriormente se modeló en la herramienta de simulación OMNeT++, los módulos creados se agrupan en un módulo compuesto (*edgeCore*) y se ubican en la posición de cada nodo predefinidos en el mapa de Estados

Unidos y de esa manera formar la topología de la red. La red modelada en la herramienta se muestra en la figura 3.10.

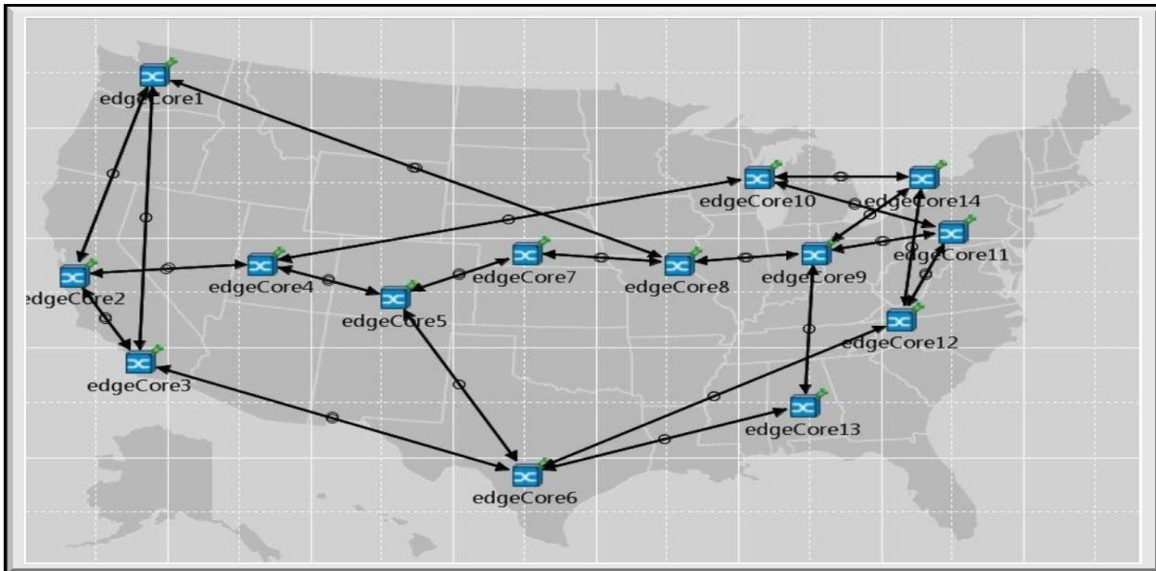


Figura 3. 10 Red OBS basada en la topología NSFNeT –implementada en OMNeT

3. 4 DIAGRAMA DE FLUJO DEL ALGORITMO DE ENSAMBLE MIXTO Y DEL DMBAА-FLPSO Y PSEUDOCÓDIGO DEL DMBAА-FLPSO

Como se mencionó en el Capítulo 2, en la sesión 2.2 y 2.3, para las redes OBS se propone un algoritmo mixto basado en la integración de la metaheurística PSO (Optimización por Enjambre de Partículas) junto con la técnica FL (Lógica Difusa) para realizar el proceso de ensamble de ráfagas de una forma adaptativa, llamado **DMBAА-FLPSO** (*Dynamic Mixed Burst Assembly Algorithms–Fuzzy Logic and PSO*).

Con base a la descripción hecha en el Capítulo 1 –sesión 1.2.3, de cada uno de los módulos del proceso de ensamble de ráfagas y a la descripción en el Capítulo 3 –sesión 3.3, de la implementación de los módulos y submódulos en cada uno de los nodos de una red OBS, se presenta el diagrama de flujo del proceso de ensamble de ráfagas con el algoritmo de ensamblaje mixto (algoritmo de ensamble clásico) y el diagrama de flujo del algoritmo de ensamble propuesto **DMBAА-FLPSO**, incluyendo su **pseudocódigo**. Los cuales, cada diagrama de flujo, se toman como base para la implementación del módulo con el algoritmo de ensamble clásico y el algoritmo de ensamble propuesto en la herramienta IDE OMNeT++.

A continuación, en la figura 3.11, se muestra el diagrama de flujo general del proceso de ensamble de ráfagas, al implementar el algoritmo de ensamble mixto clásico. Mixto, en cuanto a que implementa los dos parámetros de ensamble.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

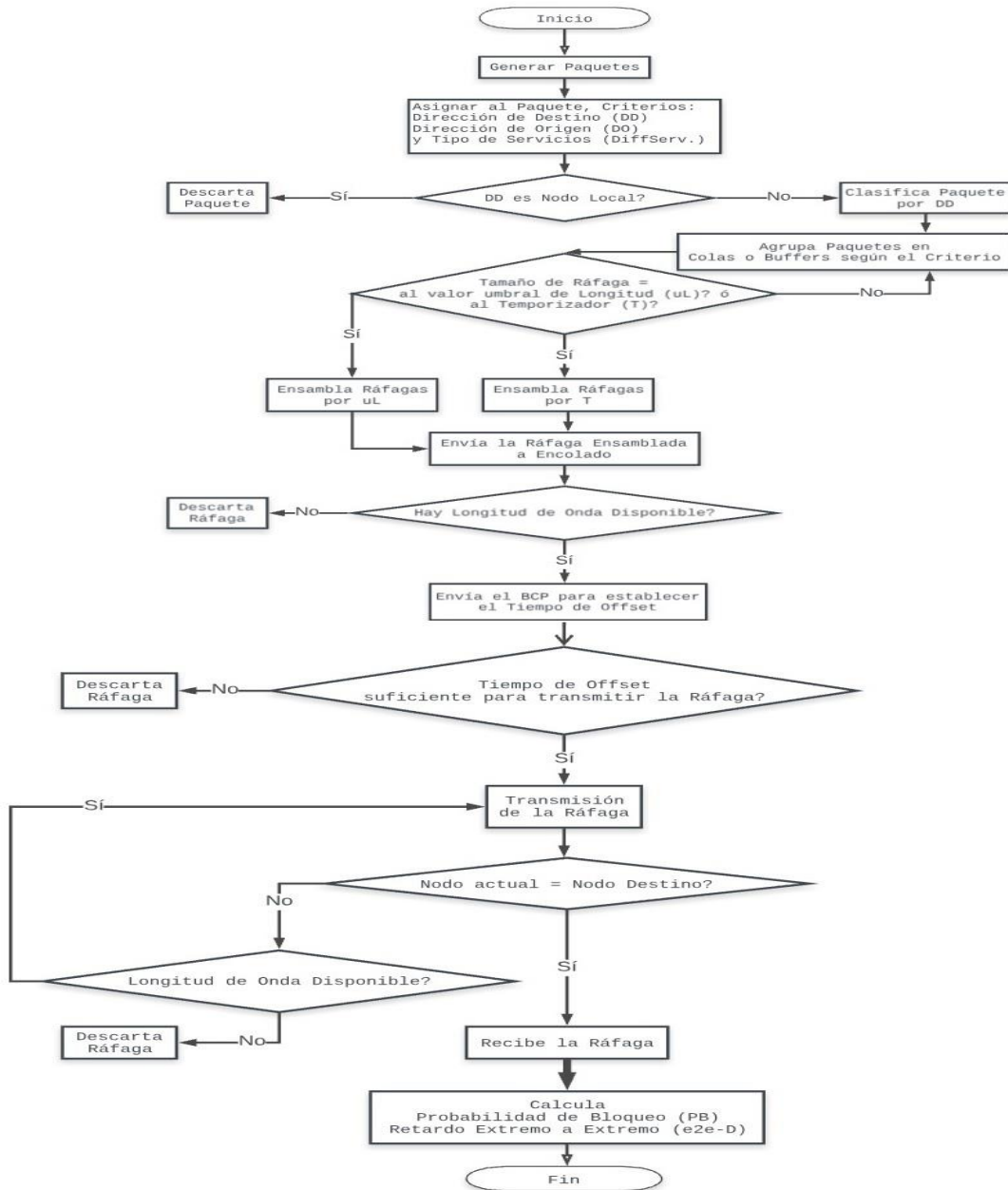


Figura 3. 11 Diagrama de Flujo General del Algoritmo de Ensamble Mixto –Clásico

El diagrama de flujo del algoritmo de ensamble **DMBAA-FLPSO**, sirve como guía para integrar la técnica Lógica Difusa (FL) y la Metaheurística PSO, con característica adaptativa, para realizar el proceso de ensamblaje y de esta manera realizar su implementación en el algoritmo a desarrollar (propuesto), en la red OBS-NSFNet, sobre el software de simulación OMNeT++. A continuación, de la figura 3.12 a la figura 3.15, se muestra el diagrama de flujo del algoritmo **DMBAA-FLPSO**, el cual, presenta cada uno de los parámetros y criterios seleccionados,



para el algoritmo de ensamble propuesto, que actúa en los tres módulos del proceso de ensamble. Destacando el módulo de Ensamblaje (Módulo 2).

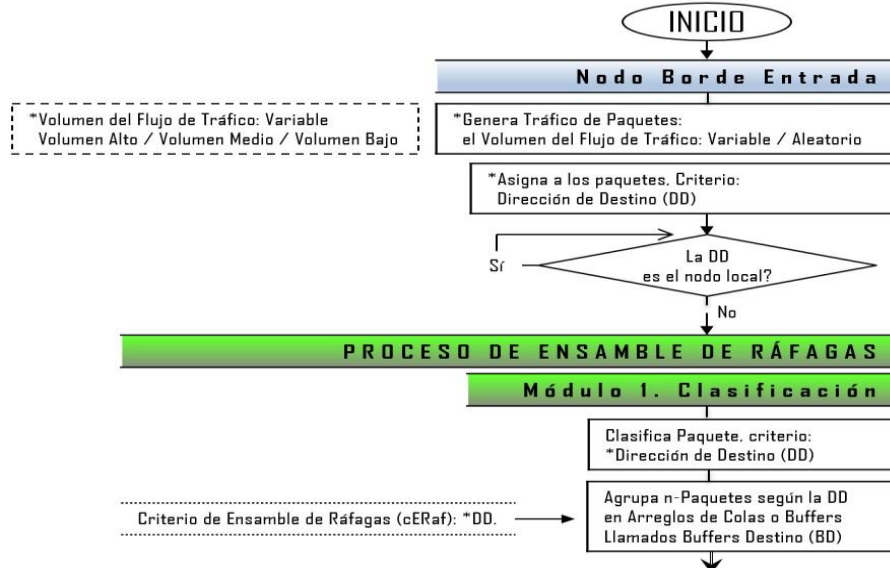


Figura 3. 12 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Nodo de Borde de Entrada OBS y Módulo 1 Clasificación

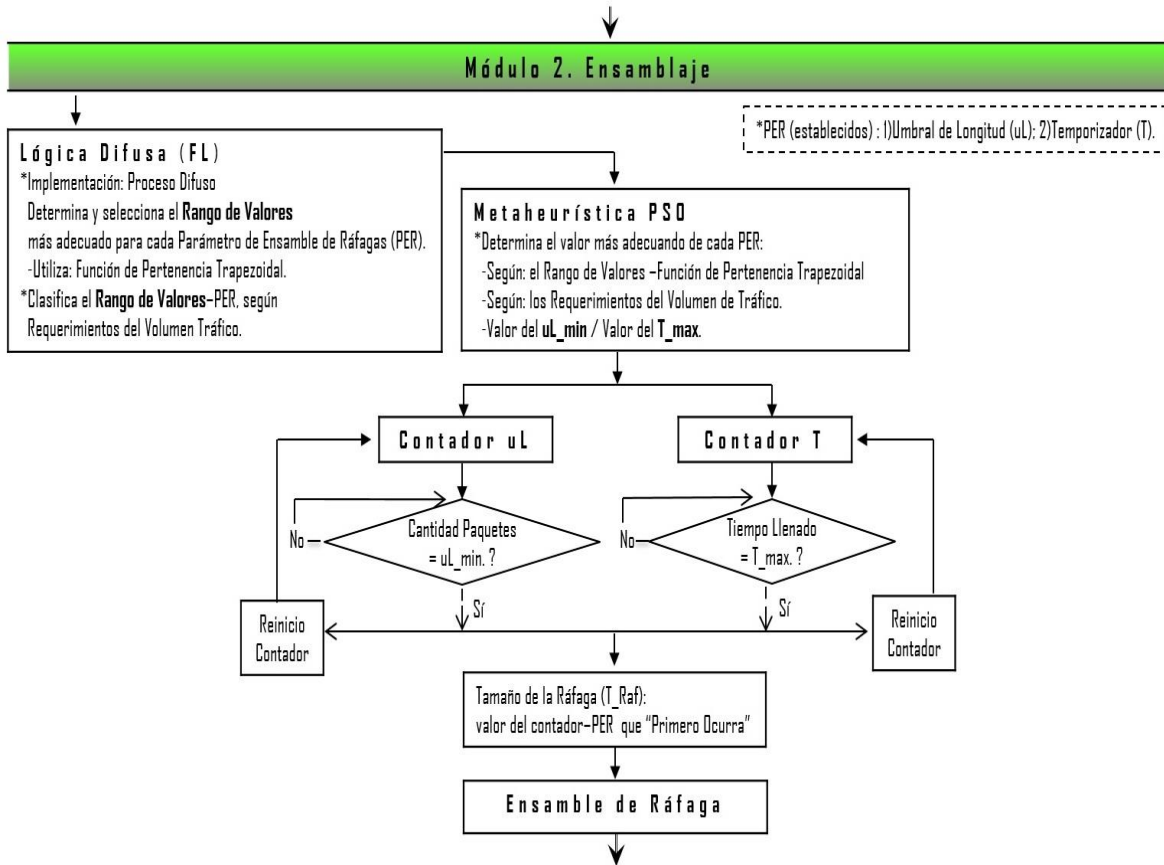


Figura 3. 13 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Módulo 2 Ensamblaje



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

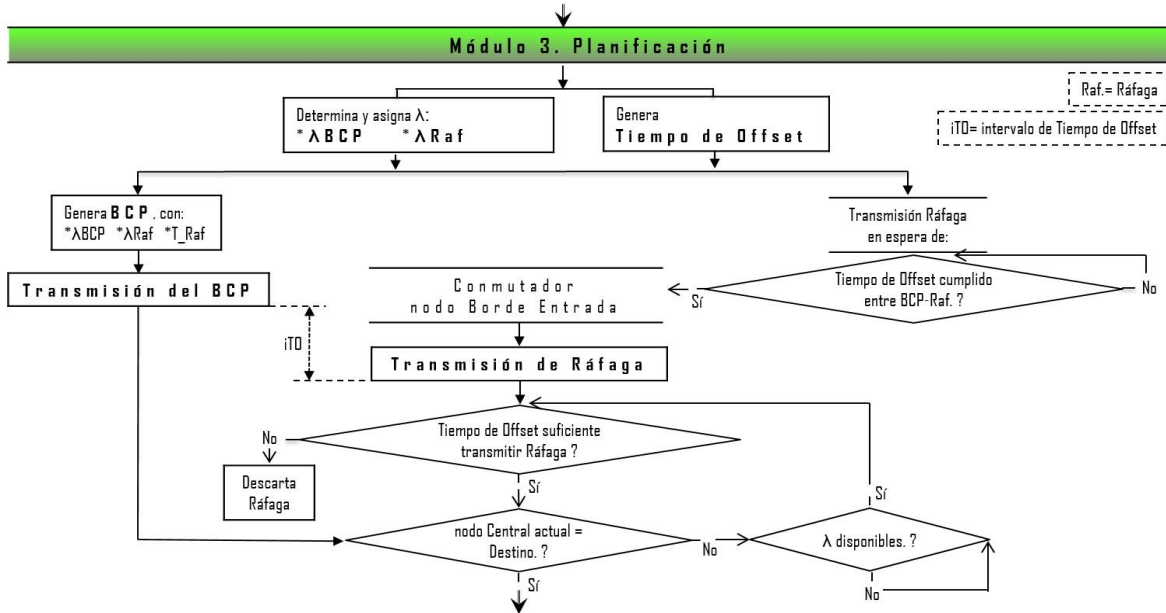


Figura 3. 14 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Módulo 3 Planificación

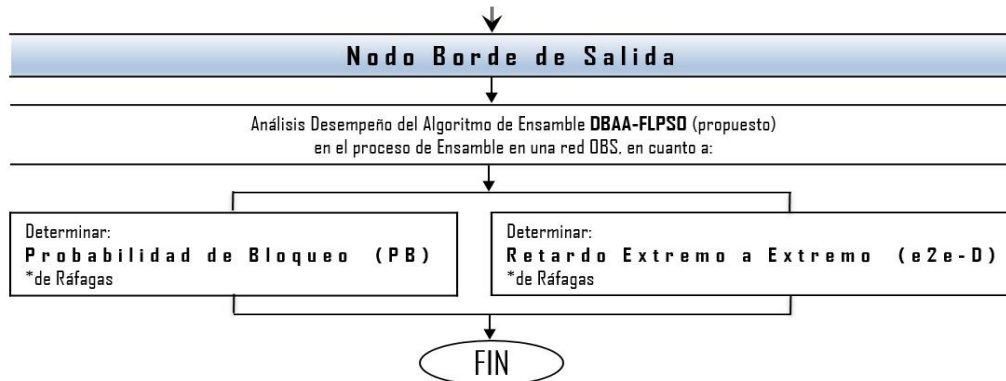


Figura 3. 15 Diagrama de Flujo del Algoritmo de Ensamble DMBAA-FLPSO. Nodo Borde de Salida OBS

A continuación, se presenta el pseudocódigo del módulo clave del algoritmo de ensamble propuesto, DMBAA-FLPSO, Módulo 2. Ensamblaje, dado que se implementa la integración de la FL y la PSO para realizar el proceso de ensamble de forma dinámica, se presenta dos esquemas:

DMBAA-FLPSO ()

- % p: Paquete; T_Raf: Tamaño de la Ráfaga.
- % BD: Cola o Buffer Destino; uL: umbral de longitud; T: tiempo de ensamble.
- % DD: Dirección de Destino (criterio de ensamble en el Buffer).
- % cP: Clasificación del Paquete según la DD;
- % PER: Parámetro de Ensamble de Ráfagas.
- % Nd: cantidad de valores en el rango de cada PER.
- %Ni_uL: cantidad de valores en el rango del uL.



% Ni_T: cantidad de valores en el rango del T.
% Volumen de flujo de tráfico entrante, se establece: Volumen Alto; Volumen Medio; Volumen Bajo.
% rangodifuso_uL[v]: se define según el volumen del flujo de tráfico entrante, se tiene: (1) rangodifuso_uL para volumen Alto; (2) rangodifuso_uL para volumen Medio; (3) rangodifuso_uL para volumen Bajo.
% rangodifuso_T[v]: se define según el volumen del flujo de tráfico entrante, se tiene: (1) rangodifuso_T para volumen Alto; (2) rangodifuso_T para volumen Meido; (3) rangodifuso_T para volumen Bajo.
% rangoinicial_uL: rango conformado por todos los posibles valores que puede tomar uL.
% rangoinicial_T: rango conformado por todos los posibles valores que puede tomar T.
% rangoinicial_T:
% rangodifuso-ini_uL: rango de valores de uL inicial. Estado Difuso.
% rangodifuso-ini_T: rango de valores de T inicial. Estado Difuso.
% rangodifuso_uL: rango de valores de uL. Es un Subconjunto del Estado Difuso.
% rangodifuso_T: rango de valores de T. Es un Subconjunto del Estado Difuso.
% $\mu_A# ()$: es la función de pertenencia trapezoidal calculada por la suma de tres ecuaciones lineales. Se aplica la función de pertenencia trapezoidal según el estado difuso, puede ser en el centro o izquierda o derecha.
% (a, b): conjunto de valores de a hasta b.

Esquema 1:

Módulo 2. Ensamblaje

% implementación de la técnica FL

funciónPertenenciaTrapezoidal ()

Para rangodifuso-ini_uL & rangodifuso-ini_T **hacer**

rangodifuso_uL[v] = rango ((a,b), μ_A (centro o izquierda o derecha))

rangodifuso_T[v] = rango ((a,b), μ_A (centro o izquierda o derecha))

% implementación de la Metaheurística PSO

nd=1

Nd= cantidad (rangodifuso_uL) = cantidad (rangodifuso_T)

v_uLmin= 1000000.

v_Tmin = 1000000.

Para uL \in rangodifuso_uL[v] & T \in rangodifuso_T[v] **hacer**

Mientras nd <= Nd **hacer**

v_uL = valormin (rangodifuso_uL[nd], v_uLmin)

v_T = valormin (rangodifuso_T[nd], v_Tmin)

v_uLmin = v_uL

v_Tmin = v_T

nd +1



**%Contador por Parámetro de Ensamble para determinar el Tamaño Ráfaga
Contador()**

```
Para v_uL & v_T hacer
  m=0
  Mientras m <= v_uL o m <= v_T hacer
    Si m= v_uL entonces T_Raf = m
    Si m= v_T entonces T_Raf = m
  m++
```

%Ensamblaje de la Ráfaga
Raf =Ensamblar (T_Raf, BD)

Esquema 2:

Módulo 2. Ensamblaje

```
m=0
ni=1
nd=1
nd1= 1
pbmin= 1000000.
v_uLmin= 1000000.
v_Tmin = 1000000.
```

%1era Implementación de la técnica FL y la Metaheurística PSO
%FL

funciónPertenenciaTriangular ()

```
Para uL ∈ rangoinicial_uL[v] & T ∈ rangoinicial_T[v] hacer
  rangodifusoi_uL[v] = rango (  $\mu_{TA}(\text{centrada})$  ,  $\mu_{TB}(\text{centrada})$ , ... )
  rangodifusoi_T[v] = rango (  $\mu_{TA}(\text{centrada})$  ,  $\mu_{TB}(\text{centrada})$ , ... )
```

Para uL ∈ rangodifusoi_uL[v] & T ∈ rangodifusoi_T[v] hacer

Ni_uL= cantidad (rangodifusoi_uL)

Ni_T= cantidad (rangodifusoi_T)

Mientras ni <= Ni hacer

```
  uL= rangodifusoi_uL[ni]
  T= rangodifusoi_T[ni]
  T_Raf= Contador()
  RafP= Ensamblar (T_Raf, BD)
  pb[ni]= PB (RafP_recibidas, RafP_perdidas, ...)
  n+1
```

%PSO

Para pb[v] hacer

Nd_uL= cantidad (rangodifusoi_uL)

Nd_v= cantidad (rangodifusoi_T)



```
Mientras nd <= Nd hacer
    Pbmin= valormin (pb[nd], pbmin)
    nd +1
v_uL = valormin (pb[v], Pbmin)
v_T = valormin (pb[v], Pbmin)

% 2da Implementación de la técnica FL y la Metaheurística PSO
%t.FL
funciónPertenenciaTrapezoidal ( )
    Para rangodifusoi_uL & v_uL & rangodifusoi_T & v_T hacer
        rangodifuso_uL[v] = rango ( (a,b),  $\mu_A$ (centrado en v_uL, rangodifusoi_uL) )
        rangodifuso_T[v] = rango ( (a,b),  $\mu_A$ (centrado en v_T, rangodifusoi_T) )

%PSO
Nd1= cantidad (rangodifuso_uL) = cantidad (rangodifuso_T)
Para uL  $\in$  rangodifuso_uL[v] & T  $\in$  rangodifuso_T[v] hacer
    Mientras nd1 <= Nd1 hacer
        v_uL = valormin (rangodifuso_uL[nd], rangodifuso_uL[nd-1])
        v_T = valormin (rangodifuso_T[nd], rangodifuso_T[nd-1])
        n+1

%Contador por Parámetro de Ensamble para determinar el Tamaño Ráfaga
Contador()
Para v_uL & v_T hacer
    Mientras m <= v_uL o m <= v_T hacer
        Si m= v_uL entonces T_Raf = m
        Si m= v_T entonces T_Raf = m
        m++

%Ensamblaje de la Ráfaga
Raf =Ensamblar (T_Raf, BD)
```



CAPÍTULO 4. ESCENARIOS Y CASOS DE SIMULACIÓN, PRUEBAS PARA LA EVALUACIÓN Y ANÁLISIS DEL DESEMPEÑO DEL ALGORITMO DMBA-FLPSO CON EL ALGORITMO DE ENSAMBLE MIXTO

4.1 INTRODUCCIÓN

En este capítulo se describen los escenarios de simulación definidos para el estudio del desempeño del algoritmo de ensamble DMBA-FLPSO en una red OBS-NSFNeT distribuida en términos de la probabilidad de bloqueo de ráfagas ($P_{B_ráfagas}$), ver ecuación 4.1 y el retardo extremo a extremo de las ráfagas transmitidas ($e_{2e-D_ráfagas}$)¹. Donde, se busca evaluar el desempeño del algoritmo propuesto para dar en cierto grado una solución al problema de ensamble de ráfagas.

$$PB = \frac{\text{Número_de_Ráfagas_Perdidas}}{\text{Número_de_Ráfagas_Totales}} \quad (4.1)$$

Un escenario de simulación hace referencia a la configuración inicial de las características y parámetros de la simulación. En este caso, se caracteriza la red OBS-NSFNeT distribuida, en la cual, se configura algunas especificaciones como el algoritmo de ensamble mixto (clásico) o el DMBA-FLPSO (propuesto), rangos de valores para cada parámetro de ensamble (PER), semilla inicial generadora de tráfico (variable), relación entre la cantidad de paquetes y ráfagas generadas según el volumen del flujo de tráfico (carga alta, carga media y carga baja). Como se presenta la tabla 4.1.

4.2 CASOS DE SIMULACIÓN

Para el estudio de este trabajo de grado se generaron dos casos de estudio de simulación con el fin de analizar el desempeño del algoritmo de ensamble mixto híbrido y adaptativo DMBA-FLPSO (propuesto), caso de estudio 2, con el algoritmo de ensamble mixto (clásico), caso de estudio 1, sobre la misma topología de red óptica NSFNeT Distribuida.

4.2.1 Caso de Estudio 1: Red OBS-NSFNeT Distribuida con el Algoritmo de Ensamble de Ráfagas Mixto (clásico)

Este caso, se caracteriza una red óptica OBS soportada sobre WDM con topología NSFNeT Distribuida (red OBS de prueba), implementando el algoritmo de

¹ El retardo extremo a extremo de ráfagas es el tiempo que tarda la ráfaga desde que es ensamblada en el nodo origen hasta el momento en que llega al desensamblador del nodo destino.



ensamble mixto (clásico) para el proceso de ensamble de ráfagas de datos. Mixto en cuanto a que establece los dos parámetros de ensamble, umbral de longitud (uL) y el tiempo de ensamblaje (T), donde, se utiliza el parámetro que primero ocurra.

4.2. 2 Caso de Estudio 2: Red OBS–NSFNeT Distribuida con el Algoritmo de Ensamble de Ráfagas Mixto Híbrido Adaptativo (propuesto)

Este caso, se caracteriza una red óptica OBS soportada sobre WDM con topología NSFNeT Distribuida (red OBS de prueba), implementando el algoritmo de ensamble mixto híbrido adaptativo DMBAA-FLPSO (propuesto) para el proceso de ensamble de ráfagas de datos. Mixto en cuanto a que establece los dos parámetros de ensamble, umbral de longitud (uL) y el tiempo de ensamblaje (T), donde, se utiliza el parámetro que primero ocurra y es híbrido dado que integra la técnica lógica difusa (FL) y la metaheurística PSO para realizar de una forma adaptativa el proceso dicho proceso.

4.2. 3 Sub Casos de Simulación

En cada caso de estudio, caso de estudio 1 y caso de estudio 2, en la red OBS de prueba cada sub caso se configura de forma fija con 2 longitudes de onda (λ) por enlace de fibra óptica y con una velocidad por enlace de 2.5 Gbps, basándose en la recomendación G.984.1 del 2003 de la ITU-T, en [104]. Por otro lado, se varía la cantidad de paquetes ensamblados en una ráfaga (uL), para los sub casos de primer orden (1, 2, 3, 4, 5) y se varia el valor del tiempo de ensamblaje o temporizador (T), para los sub casos de segundo orden (1.1, 2.1, 3.1, 4.1, 5.1, 1.2, 2.2, 3.2, 4.2, 5.2), con el fin de analizar el desempeño del algoritmo DMBAA-FLPSO al compararlo con el algoritmo de ensamble mixto. El desempeño en cuanto a la PB y el e2e-D.

Sub Caso 1

Para el sub caso 1, de forma específica, la red OBS de prueba se configura el parámetro de ensamble umbral de longitud (uL) con un valor de 2 paquetes por ráfaga, como se especifica en la tabla 4.1.

Sub Caso 2

Para el sub caso 2, de forma específica, la red OBS de prueba se configura el parámetro de ensamble umbral de longitud (uL) con un valor de 3 paquetes por ráfaga, como se especifica en la tabla 4.1.



Tabla 4. 1 Escenario y Casos de Estudio de Simulación

ESCENARIOS DE SIMULACIÓN									
Escenario: Topología de red OBS-NSFNet Distribuida									
Casos de Simulación									
Caso de Estudio 1 Implementando Algoritmo de Ensamble Mixto (Clásico).					Caso de Estudio 2 Implementando Algoritmo de Ensamble Mixto Adaptativo basado en la integración de la técnica Lógica Difusa y la Metaheurística PSO (híbrido), DMBAA-FLPSO.				
Sub Casos de Simulación									
Rangos-Parámetros de Ensamble de Ráfagas (con enlace WDM: 2 λ)									
Sub Caso 1 (uL números de paquetes) 2		Sub Caso 2 (uL números de paquetes) 3		Sub Caso 3 (uL números de paquetes) 4		Sub Caso 4 (uL números de paquetes) 5		Sub Caso 5 (uL números de paquetes) 6	
Sub Caso (T mili-segundos) 1.1 1.2 1.3 2.25 2.5 2.75		Sub Caso (T mili-segundos) 2.1 2.2 2.3 2.25 2.5 2.75		Sub Caso (T mili-segundos) 3.1 3.2 3.3 2.25 2.5 2.75		Sub Caso (T mili-segundos) 4.1 4.2 4.3 2.25 2.5 2.75		Sub Caso (T mili-segundos) 5.1 5.2 5.3 2.25 2.5 2.75	
Relación entre los Sub Casos de Simulación									
el Volumen del Flujo de Tráfico varía según la Semilla inicial Generadora de Tráfico y el Volumen del Flujo de Tráfico esta dado según el PER uL (Carga en Erlang / Semilla-valor inicial / uL es número de paquetes por ráfaga)									
Muy Alta	Muy Alta	Alta	Alta	Media	Media	Baja	Baja	Muy Baja	Muy Baja
20	70	150	532570	297834	33	7385	40187	9	108
4, 5, 6	4, 5, 6	4, 5, 6	4, 5, 6	3, 4, 5	3, 4, 5	2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4



Sub Caso 3

Para el sub caso 3, de forma específica, la red OBS de prueba se configura el parámetro de ensamble umbra de longitud (uL) con un valor de 4 paquetes por ráfaga, como se especifica en la tabla 4.1.

Sub Caso 4

Para el sub caso 1, de forma específica, la red OBS de prueba se configura el parámetro de ensamble umbra de longitud (uL) con un valor de 5 paquetes por ráfaga, como se especifica en la tabla 4.1.

Sub Caso 5

Para el sub caso 1, de forma específica, la red OBS de prueba se configura el parámetro de ensamble umbra de longitud (uL) con un valor de 6 paquetes por ráfaga, como se especifica en la tabla 4.1.

Sub Casos 1.1, 2.1, 3.1, 4.1 y 5.1

Se configuran las dos redes OBS de prueba, red del caso de estudio 1 y la red del caso de estudio 2, con el parámetro tiempo de ensamblaje (T) con un valor de 0.00225 segundos (2.25ms), como se especifica en la tabla 4.1.

Sub Casos 1.2, 2.2, 3.2, 4.2 y 5.2

Se configuran las dos redes OBS de prueba, red del caso de estudio 1 y la red del caso de estudio 2, con el parámetro de ensamble el tiempo de ensamblaje (T) con un valor de 2.25 ms (0.00225 segundos), como se especifica en la tabla 4.1.

Sub Casos 1.3, 2.3, 3.3, 4.3 y 5.3

Se configuran las dos redes OBS de prueba, red del caso de estudio 1 y la red del caso de estudio 2, con el parámetro de ensamble el tiempo de ensamblaje (T) con un valor de 2.75 ms (0.00275 segundos), como se especifica en la tabla 4.1

4. 3 PRUEBAS DE SIMULACIÓN

Dado que el objetivo principal de este trabajo de grado de maestría es determinar el valor adecuado de cada parámetro de ensamble, teniendo en cuenta los requerimientos que exige en cierto instante la carga de tráfico, volumen del flujo de tráfico variable: alto, medio y bajo, para asignar el tamaño óptimo a la ráfaga a transmitir y de esta manera evaluar de qué forma influye en el desempeño de toda la red OBS de prueba, se utiliza la técnica lógica difusa (FL), primera parte del algoritmo propuesto, para establecer un conjunto de valores (conjunto difuso) para cada parámetro de ensamble, el cual, depende de una función de pertenencia y de

la carga de tráfico. Para esto, como se muestra en la tabla 4.1, la carga de tráfico se clasifica por niveles de carga, dados en Erlang (E), a cada nivel se le asigna una semilla inicial generadora de tráfico diferente², resultando los siguiente 10 niveles: (1) para **carga Muy Alta** con Semilla1= 20; (2) para **carga Muy Alta** con Semilla2= 70; (3) para **carga Alta** con Semilla3= 150; (4) para **carga Alta** con Semilla4= 532570; (5) para **carga Media** con Semilla5= 297834; (6) para **carga Media** con Semilla6= 33; (7) para **carga Baja** con Semilla7= 7385; (8) para **carga Baja** con semilla8= 40187; (9) para **carga Muy Baja** con Semilla9= 9; (10) para **carga Muy Baja** con Semilla10= 108.

Además, se realiza un análisis de los posibles valores que puede tomar los dos parámetros de ensamble utilizando la FL, con la función de pertenencia Trapezoidal y con la función de pertenencia Triangular, ver figura 4.1 hasta la 4.4, como resultado se fijan los siguientes conjuntos difusos para cada parámetro teniendo en cuenta el nivel de carga de tráfico: (1) para **carga Muy Alta** y **carga Alta** se utiliza el conjunto difuso para $u_L=(4, 5, 6)$ y para $T=(2.25, 2.5, 2.75)$; (2) para **carga Media** se utiliza el conjunto de difuso para $u_L=(3, 4, 5)$ y para $T=(2.25, 2.5, 2.75)$; (3) para **carga Baja** y **carga Muy Baja** se utiliza el conjunto difuso para $u_L=(2, 3, 4)$ y para $T=(2.25, 2.5, 2.75)$. En la tabla 4.1, se presenta la relación entre los niveles de carga según el conjunto difuso de u_L .

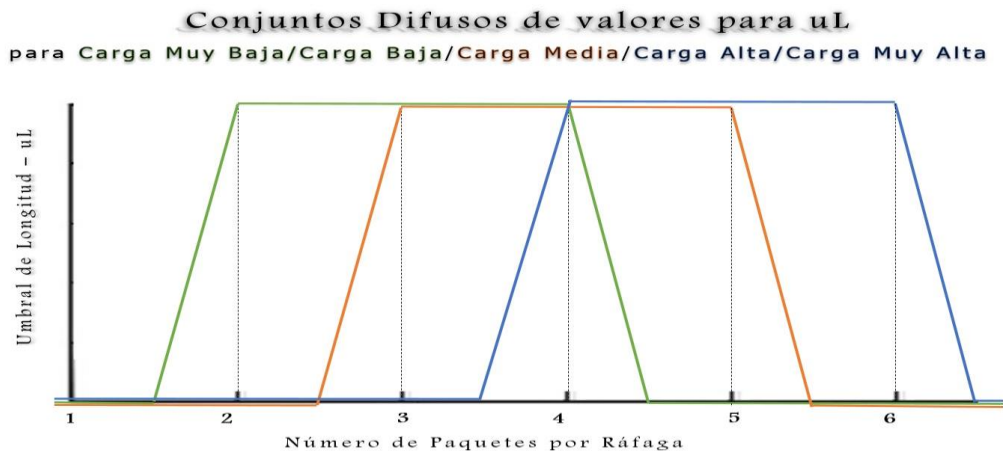


Figura 4. 1 Conjuntos Difusos u_L utilizando la técnica Lógica Difusa con Función Trapezoidal para Carga Muy Baja, Carga Baja, Carga Media, Carga Alta y Carga Muy Alta

Con base a la tabla 4.1, en la tabla 4.2 hasta la tabla 4.11 se presentan las diferentes combinaciones para establecer las pruebas de simulación y en la figura 4.5 hasta la figura 4.9 se presentan las pruebas de simulación a realizar por la técnica Lógica Difusa (FL) en el algoritmo de ensamble propuesto DMBA-FLPSO. La FL, implementa la función de pertenencia Trapezoidal para determinar los conjuntos difusos según el nivel de carga de tráfico e implementa la función de pertenencia Triangular para determinar el valor de la PB y del e2e-D de cada elemento dentro de los conjuntos difusos, ver figura 4.1 hasta la 4.4. El proceso

² El objetivo de variar la semilla es trabajar con un flujo de tráfico aleatorio-estocástico sobre la red.



final del DMBAA-FLPSO, consiste en utilizar la PSO para seleccionar el valor adecuado de la PB y del e2e-D de cada conjunto difuso, realizando una comparación y seleccionando el menor valor de cada conjunto.

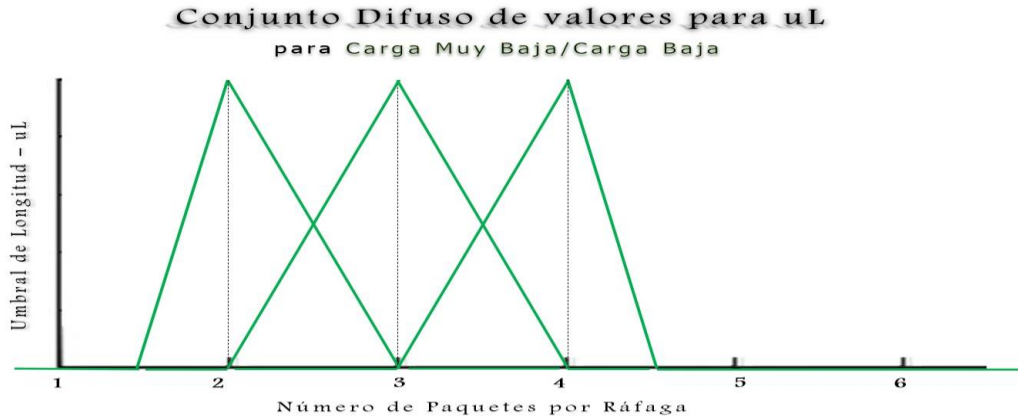


Figura 4. 2 Valores uL utilizando la técnica Lógica Difusa con Función Triangular para Carga Muy Baja y Carga Baja

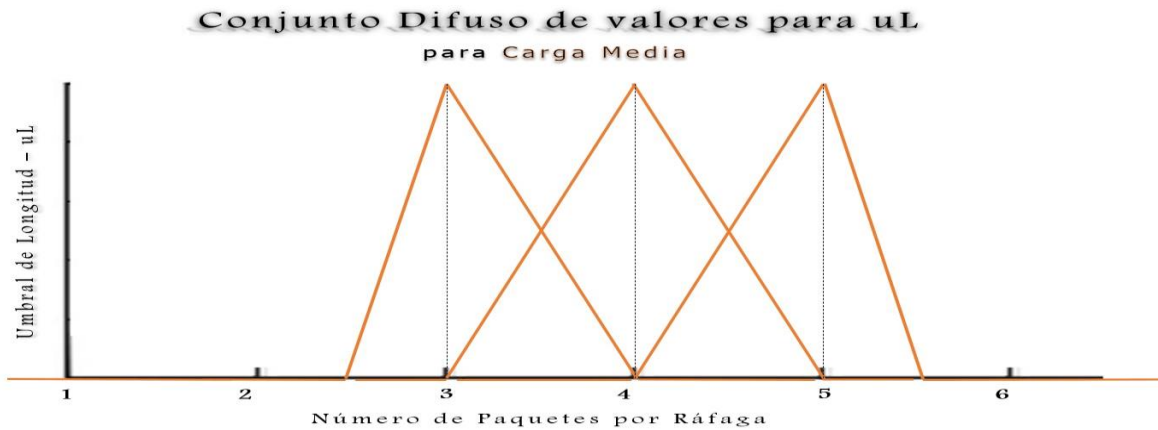


Figura 4. 3 Valores uL utilizando la técnica Lógica Difusa con Función Triangular para Carga Muy Media

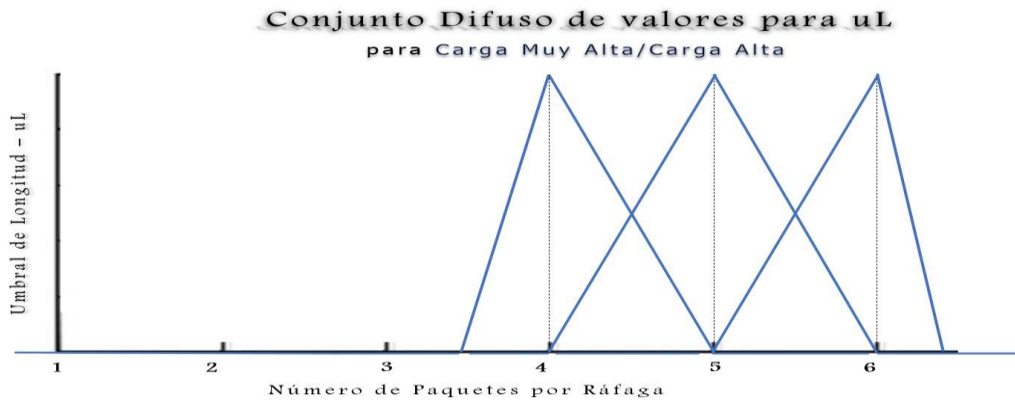


Figura 4. 4 Valores uL utilizando la técnica Lógica Difusa con Función Triangular para Carga Muy Alta y Carga Alta



Tabla 4. 2 Pruebas de Simulación para Carga Muy Alta con Semilla 20, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.

Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 20.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 20.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3. Con Semilla 20.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.1. Con Semilla 20.	$\begin{cases} uL = 5 \\ T = 2.25ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.2. Con Semilla 20.	$\begin{cases} uL = 5 \\ T = 2.5ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.3. Con Semilla 20.	$\begin{cases} uL = 5 \\ T = 2.75ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.1. Con Semilla 20.	$\begin{cases} uL = 6 \\ T = 2.25ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.2. Con Semilla 20.	$\begin{cases} uL = 6 \\ T = 2.5ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.3 Con Semilla 20.	$\begin{cases} uL = 6 \\ T = 2.75ms \end{cases}$

Tabla 4. 3 Pruebas de Simulación para Carga Muy Alta con Semilla 70, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.

Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 70.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 70.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3. Con Semilla 70.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.1. Con Semilla 70.	$\begin{cases} uL = 5 \\ T = 2.25ms \end{cases}$



Para Sub Caso 5 con Sub Caso 5.2. Con Semilla 70.	$\begin{cases} uL = 5 \\ T = 2.5ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.3. Con Semilla 70.	$\begin{cases} uL = 5 \\ T = 2.75ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.1. Con Semilla 70.	$\begin{cases} uL = 6 \\ T = 2.25ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.2. Con Semilla 70.	$\begin{cases} uL = 6 \\ T = 2.5ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.3 Con Semilla 70.	$\begin{cases} uL = 6 \\ T = 2.75ms \end{cases}$

Tabla 4. 4 Pruebas de Simulación para Carga Alta con Semilla 150, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.

Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 150.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 150.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3. Con Semilla 150.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.1. Con Semilla 150.	$\begin{cases} uL = 5 \\ T = 2.25ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.2. Con Semilla 150.	$\begin{cases} uL = 5 \\ T = 2.5ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.3. Con Semilla 150.	$\begin{cases} uL = 5 \\ T = 2.75ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.1. Con Semilla 150.	$\begin{cases} uL = 6 \\ T = 2.25ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.2. Con Semilla 150.	$\begin{cases} uL = 6 \\ T = 2.5ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.3 Con Semilla 150.	$\begin{cases} uL = 6 \\ T = 2.75ms \end{cases}$



Tabla 4. 5 Pruebas de Simulación para Carga Alta con Semilla 532570, para la combinación del Sub Caso 4, Sub Caso 5 y Sub Caso 6 con los Sub Casos 4.1, 4.2 y 4.3, Sub Casos 5.1 5.2 y 5.3, y Sub Casos 6.1, 6.2 y 6.3 respectivamente.

Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 532570.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 532570.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3. Con Semilla 532570.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.1. Con Semilla 532570.	$\begin{cases} uL = 5 \\ T = 2.25ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.2. Con Semilla 532570.	$\begin{cases} uL = 5 \\ T = 2.5ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.3. Con Semilla 532570.	$\begin{cases} uL = 5 \\ T = 2.75ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.1. Con Semilla 532570.	$\begin{cases} uL = 6 \\ T = 2.25ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.2. Con Semilla 532570.	$\begin{cases} uL = 6 \\ T = 2.5ms \end{cases}$
Para Sub Caso 6 con Sub Caso 6.3 Con Semilla 532570.	$\begin{cases} uL = 6 \\ T = 2.75ms \end{cases}$

Tabla 4. 6 Pruebas de Simulación para Carga Media con Semilla 297834, para la combinación del Sub Caso 3, Sub Caso 4 y Sub Caso 5 con los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3, y Sub Casos 5.1, 5.2 y 5.3 respectivamente.

Para Sub Caso 3 con Sub Caso 3.1. Con Semilla 297834.	$\begin{cases} uL = 3 \\ T = 2.25ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.2. Con Semilla 297834.	$\begin{cases} uL = 3 \\ T = 2.5ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.3. Con Semilla 297834.	$\begin{cases} uL = 3 \\ T = 2.75ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 297834.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$



Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 297834.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3. Con Semilla 297834.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.1. Con Semilla 297834.	$\begin{cases} uL = 5 \\ T = 2.25ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.2. Con Semilla 297834.	$\begin{cases} uL = 5 \\ T = 2.5ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.3 Con Semilla 297834.	$\begin{cases} uL = 5 \\ T = 2.75ms \end{cases}$

Tabla 4. 7 Pruebas de Simulación para Carga Media con Semilla 33, para la combinación del Sub Caso 3, Sub Caso 4 y Sub Caso 5 con los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3, y Sub Casos 5.1, 5.2 y 5.3 respectivamente.

Para Sub Caso 3 con Sub Caso 3.1. Con Semilla 33.	$\begin{cases} uL = 3 \\ T = 2.25ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.2. Con Semilla 33.	$\begin{cases} uL = 3 \\ T = 2.5ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.3. Con Semilla 33.	$\begin{cases} uL = 3 \\ T = 2.75ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 33.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 33.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3. Con Semilla 33.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.1. Con Semilla 33.	$\begin{cases} uL = 5 \\ T = 2.25ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.2. Con Semilla 33.	$\begin{cases} uL = 5 \\ T = 2.5ms \end{cases}$
Para Sub Caso 5 con Sub Caso 5.3 Con Semilla 33.	$\begin{cases} uL = 5 \\ T = 2.75ms \end{cases}$



Tabla 4. 8 Pruebas de Simulación para Carga Baja con Semilla 7385, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.

Para Sub Caso 2 con Sub Caso 2.1. Con Semilla 7385.	$\begin{cases} uL = 2 \\ T = 2.25ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.2. Con Semilla 7385.	$\begin{cases} uL = 2 \\ T = 2.5ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.3. Con Semilla 7385.	$\begin{cases} uL = 2 \\ T = 2.75ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.1. Con Semilla 7385.	$\begin{cases} uL = 3 \\ T = 2.25ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.2. Con Semilla 7385.	$\begin{cases} uL = 3 \\ T = 2.5ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.3. Con Semilla 7385.	$\begin{cases} uL = 3 \\ T = 2.75ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 7385.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 7385.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3 Con Semilla 7385.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$

Tabla 4. 9 Pruebas de Simulación para Carga Baja con Semilla 40187, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.

Para Sub Caso 2 con Sub Caso 2.1. Con Semilla 40187.	$\begin{cases} uL = 2 \\ T = 2.25ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.2. Con Semilla 40187.	$\begin{cases} uL = 2 \\ T = 2.5ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.3. Con Semilla 40187.	$\begin{cases} uL = 2 \\ T = 2.75ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.1. Con Semilla 40187..	$\begin{cases} uL = 3 \\ T = 2.25ms \end{cases}$



Para Sub Caso 3 con Sub Caso 3.2. Con Semilla 40187.	$\begin{cases} uL = 3 \\ T = 2.5ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.3. Con Semilla 40187.	$\begin{cases} uL = 3 \\ T = 2.75ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 40187.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 40187.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3 Con Semilla 40187.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$

Tabla 4. 10 Pruebas de Simulación para Carga Muy Baja con Semilla 9, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.

Para Sub Caso 2 con Sub Caso 2.1. Con Semilla 9.	$\begin{cases} uL = 2 \\ T = 2.25ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.2. Con Semilla 9	$\begin{cases} uL = 2 \\ T = 2.5ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.3. Con Semilla 9.	$\begin{cases} uL = 2 \\ T = 2.75ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.1. Con Semilla 9.	$\begin{cases} uL = 3 \\ T = 2.25ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.2. Con Semilla 9.	$\begin{cases} uL = 3 \\ T = 2.5ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.3. Con Semilla 9.	$\begin{cases} uL = 3 \\ T = 2.75ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 9.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 9.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3 Con Semilla 9.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$



Tabla 4. 11 Pruebas de Simulación para Carga Muy Baja con Semilla 108, para la combinación del Sub Caso 2, Sub Caso 3 y Sub Caso 4 con los Sub Casos 2.1, 2.2 y 2.3, los Sub Casos 3.1, 3.2 y 3.3, Sub Casos 4.1 4.2 y 4.3 respectivamente.

Para Sub Caso 2 con Sub Caso 2.1. Con Semilla 108.	$\begin{cases} uL = 2 \\ T = 2.25ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.2. Con Semilla 108.	$\begin{cases} uL = 2 \\ T = 2.5ms \end{cases}$
Para Sub Caso 2 con Sub Caso 2.3. Con Semilla 108.	$\begin{cases} uL = 2 \\ T = 2.75ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.1. Con Semilla 108.	$\begin{cases} uL = 3 \\ T = 2.25ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.2. Con Semilla 108.	$\begin{cases} uL = 3 \\ T = 2.5ms \end{cases}$
Para Sub Caso 3 con Sub Caso 3.3. Con Semilla 108.	$\begin{cases} uL = 3 \\ T = 2.75ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.1. Con Semilla 108.	$\begin{cases} uL = 4 \\ T = 2.25ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.2. Con Semilla 108.	$\begin{cases} uL = 4 \\ T = 2.5ms \end{cases}$
Para Sub Caso 4 con Sub Caso 4.3 Con Semilla 108.	$\begin{cases} uL = 4 \\ T = 2.75ms \end{cases}$

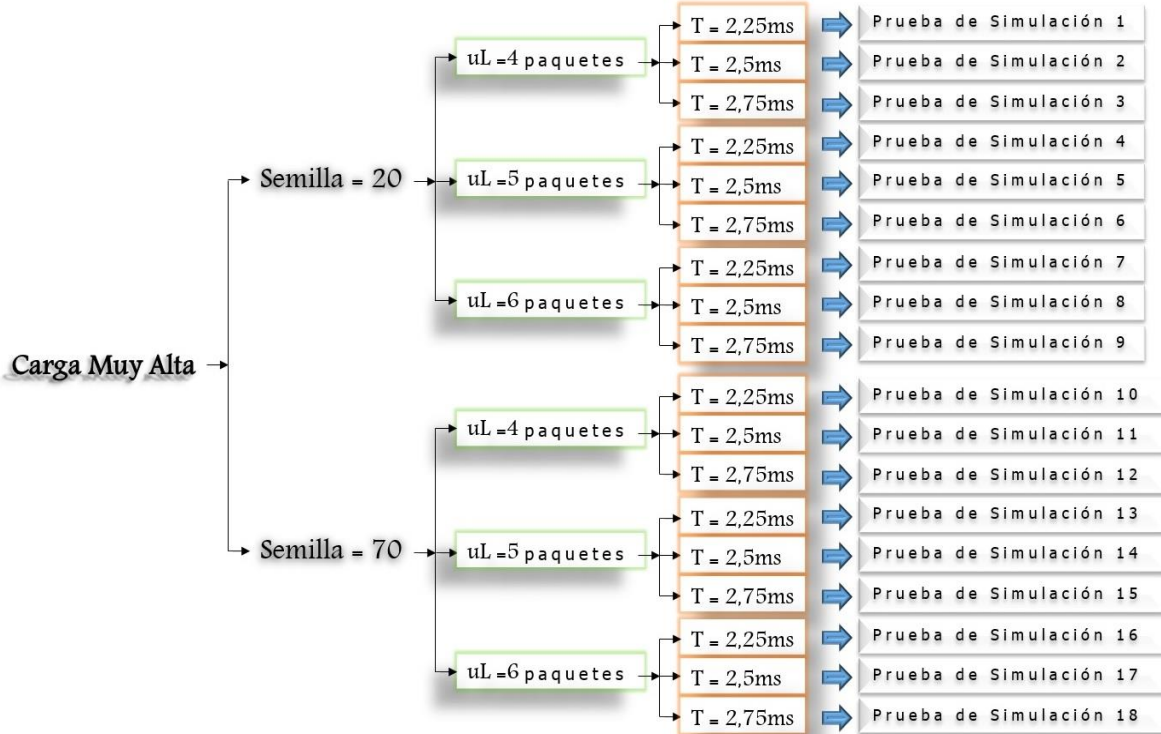


Figura 4. 5 Pruebas de Simulación para Carga Muy Alta con Semilla 20 y 70 con $uL=(4, 5, 6)$ y $T=(2.25, 2.5, 2.75)$

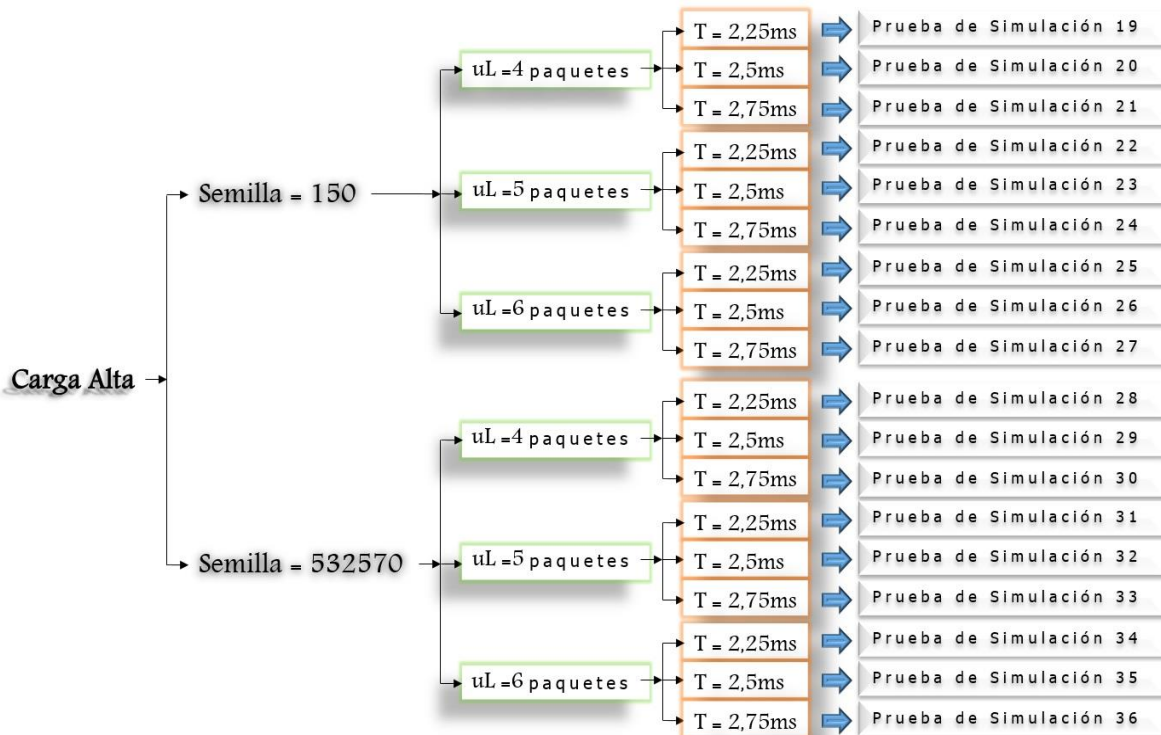


Figura 4. 6 Pruebas de Simulación para Carga Alta con Semilla 150 y 532570 con $uL=(4, 5, 6)$ con $T=(2.25, 2.5, 2.75)$

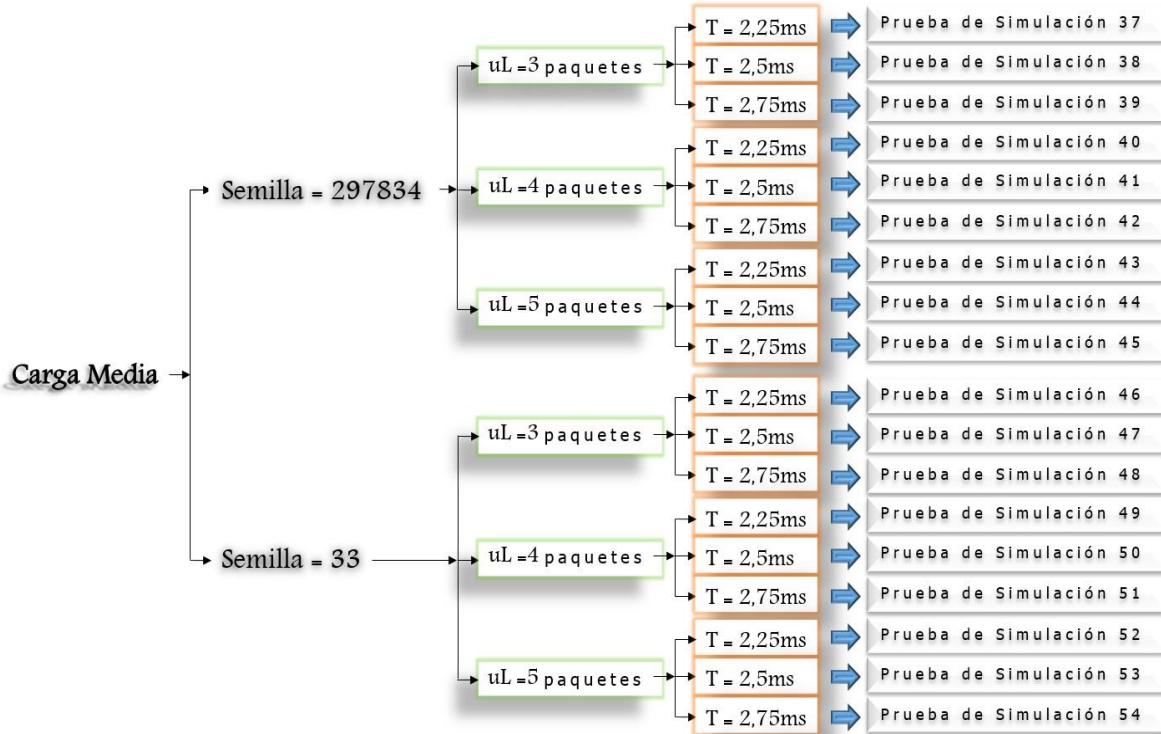


Figura 4. 7 Pruebas de Simulación para Carga Media con Semilla 297834 y 33 con $uL=(3, 4, 5)$ con $T=(2.25, 2.5, 2.75)$

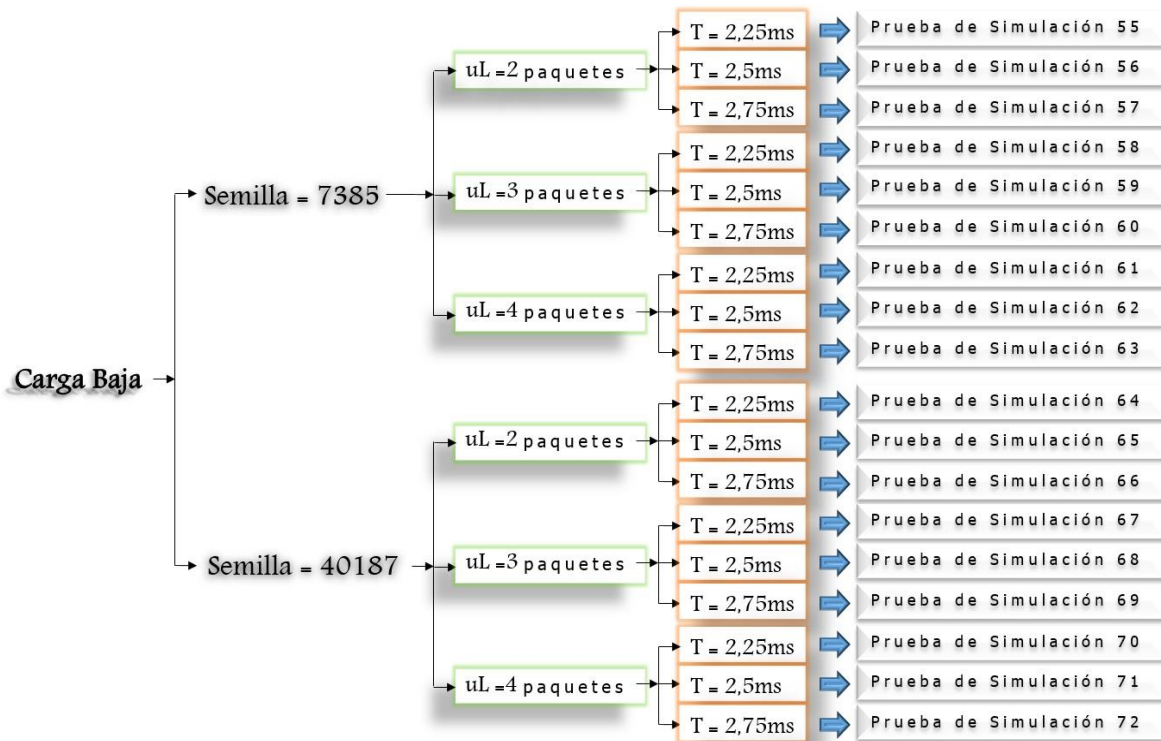


Figura 4. 8 Pruebas de Simulación para Carga Baja con Semilla 7385 y 40187 con $uL=(3, 4, 5)$ con $T=(2.25, 2.5, 2.75)$

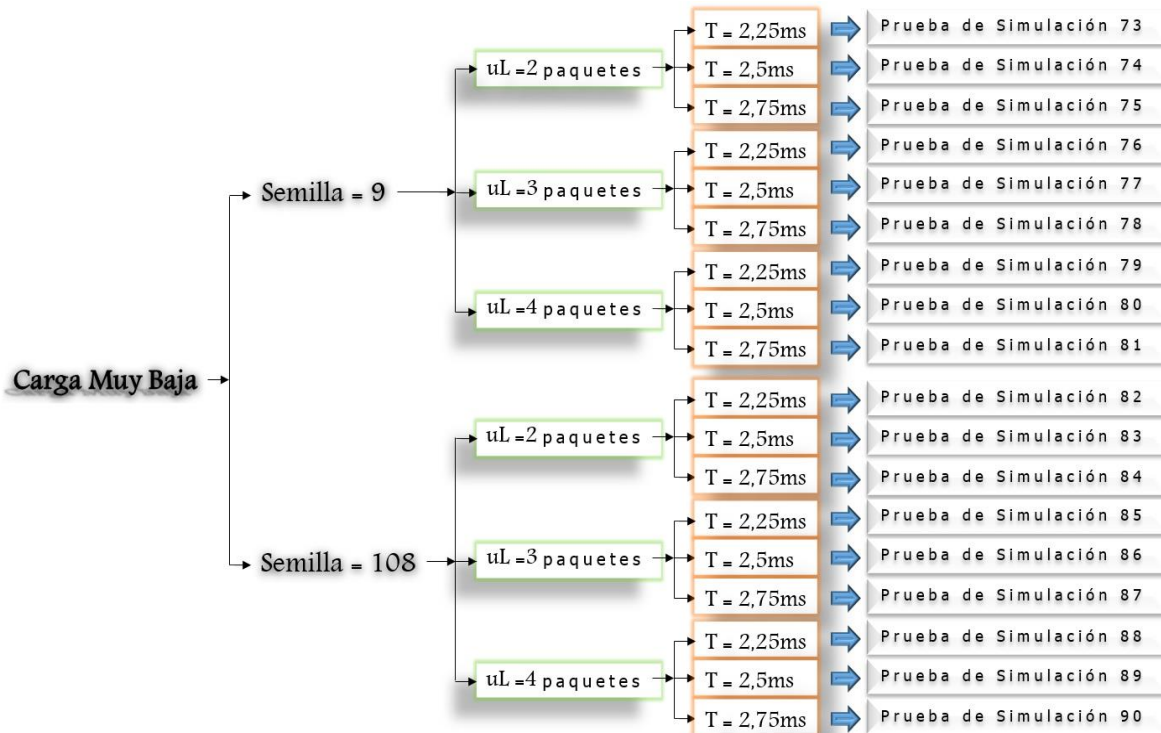


Figura 4. 9 Pruebas de Simulación para Carga Muy Baja con Semilla 9 y 108 con $uL=(2, 3, 4)$ con $T=(2.25, 2.5, 2.75)$

4. 4 PARÁMETROS A CONFIGURAR EN LOS ESCENARIOS DE SIMULACIÓN

En esta sección se identifican las variables que afectan tanto de manera directa como indirectamente el desempeño de la red y de igual manera los resultados de las simulaciones. En la tabla 4.12 se representan los parámetros configurados en la red OBS-NSFNeT Distribuida (red de prueba) para las diferentes pruebas de simulación.

Tabla 4. 12 Parámetros a configurar en la red OBS NSFNeT Distribuida

Parámetro	Nombre de la variable en OMNeT ++	Valores o intervalos
Número de longitudes de onda	lambda score	2 λ
Espaciamento entre canales	NA	50 GHZ
Velocidad de transmisión	dataRate	2,5 Gbps
Longitud de paquete	packetLength	1472 Bytes
Tiempo de procesamiento BCP	BCPProcessingDelay	10 μ s
Tiempo de procesamiento de conmutador óptico	OXCDelay	10 μ s
Tiempo de Conversión OEO	OEOConversionDelay	5 μ s



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

	EOConversionDelay	5 μ s
Tiempo de guarda	guardTime	1 ns
Tiempo offset	minOffset	20 μ s
	maxOffset	50 μ s
Tamaño mínimo de la ráfaga	minSizePadding	8 Bytes
Tamaño del BCP	BCPSize	16B
Número máximo de paquetes por ráfaga	numPackets (#)	2
		3
		4
		5
		6
Tiempo máximo de ensamble	MaxTime (en Segundos – s)	0,001-a-0,0055
		0,00225
		0,0025
		0,00275
Tiempo Generador de Tráfico (vs la Carga de Tráfico)	sendIaTime {exponencial} (en microsegundos – μ s) Vs Carga de Tráfico	5.0 μ s Carga Muy Baja _0.1E
		4.9 μ s Carga Muy Baja _0.2E
		4.7 μ s Carga Baja _0.3E
		4.6 μ s Carga Baja _0.4E
		4.4 μ s Carga Media _0.5E
		4.3 μ s Carga Media _0.6E
		4.1 μ s Carga Alta _0.7E
		4.0 μ s Carga Alta _0.8E
		3.8 μ s Carga Muy Alta _0.9E
		3.7 μ s Carga Muy Alta _1.0E
		5.0 μ s _0.2s
		4.9 μ s _0.2s
		4.7 μ s _0.4s
4.6 μ s _0.4s		
4.4 μ s _0.6s		
4.3 μ s _0.6s		
4.1 μ s _0.78s/0.8s		
4.0 μ s _0.78s/0.8s		
3.8 μ s _1.2s		
3.7 μ s _1.2s		
Tiempo Generador de Tráfico O Tiempo entre paquetes (vs Tiempo de Simulación)	sendIaTime {exponencial} (en microsegundos – μ s) Vs Tiempo de Simulación (en segundos – s)	5.0 μ s _0.2s
		4.9 μ s _0.2s
		4.7 μ s _0.4s
		4.6 μ s _0.4s
		4.4 μ s _0.6s
		4.3 μ s _0.6s
		4.1 μ s _0.78s/0.8s
		4.0 μ s _0.78s/0.8s
		3.8 μ s _1.2s
		3.7 μ s _1.2s



Por otro lado, los nodos OBS fueron configurados para generar diferentes granularidades o densidades de tráfico variando el parámetro Tiempo entre Paquetes, SendlaTime (en OMNeT++), como se muestra en la tabla 4.12, de esta manera observar el comportamiento de la red al someterse a diferentes volumen de flujos de tráfico (o niveles de carga de tráfico). El parámetro sendlaTime esta modelado por una función exponencial, donde el argumento de la función toma valores entre 3.7 μ s y 5.0 μ s, siendo el 3.7 μ s (para 1E) y el 3.8 μ s (para 0.9E) el valor para modelar la **carga Muy Alta**, 4.0 μ s (para 0.8E) y el 4.1 μ s (para 0.7E) el valor para modelar la **carga Alta**, el 4.3 μ s (para 0.6E) y el 4.4 μ s (para 0.5E) el valor para modelar la **carga Media**, 4.6 μ s (para 0.4E) y el 4.7 μ s (para 0.3E) el valor para modelar la **carga Baja** y el 4.9 μ s (para 0.2E) y 5.0 μ s (para 0.1E) el valor para modelar la **carga Muy Baja**. Estos valores son elegidos con base en pruebas realizadas y a fin que los resultados de la simulación sean aptos para realizar un buen análisis.

Para llevar a cabo las diferentes pruebas de simulación se utilizaron 4 equipos de cómputo en los que se instaló la herramienta IDE OMNeT++ versión 4.6. En la tabla 4.13, se describen las especificaciones técnicas de los equipos utilizados en la ejecución de las simulaciones.

Tabla 4. 13 Especificaciones técnicas de los equipos utilizados.

Equipo	Procesador	Memoria RAM	Disco Duro	Sistema Operativo
Toshiba S55 A5279	Intel Core i7	8 GB	1000 GB	Windows 8.1, 64 bits
DELL Optiplex 7010	Intel Core i7	8 GB	500 GB	Windows 10, 64 bits
DELL Inspiron 620s	Intel Core i5	4 GB	1000 GB	Windows 7, 64 bits
Lenovo	AMD A10	8 GB	1000 GB	Windows 10, 64 bits

4. 5 CANTIDAD TOTAL DE SIMULACIONES Y SU EJECUCIÓN

Una vez configurada la red OBS-NSFNeT Distribuida (red de prueba) con los parámetros de la tabla 4.12, se procede a ejecutar las simulaciones sobre la herramienta IDE OMNeT++, donde se realizaron 690 pruebas de simulación en total. Estas son:

- Prueba inicial
Conjunto de valores del parámetro de ensamble uL, con los siguientes valores: 1, 2, 3, 4, 5, 6. Ver gráfica de la 4.1 a la 4.4.

Para determinar los valores del conjunto del parámetro de ensamble T, se realiza un análisis inicial de 14 valores de tiempo (en mili-segundos)



utilizando el algoritmo DMBA-FLPSO, estos valores son: 5.5, **5**, 4.5, **4**, 3.75, 3.5, 3.25, **3**, 2.75, 2.5, 2.25, **2**, 1.5, **1**.

Por lo tanto, son: 14 valores de T * 3 valores de uL por carga tráfico * 10 valores de tráfico normalizado (0.1 a 1 Erlang) (o número de semillas), ver tabla 4.12, esto da 420 Pruebas iniciales de Simulación.

- Prueba parte 1 del Algoritmo DMBA-FLPSO:
Utilizando solo la técnica Lógica Difusa (FL), son 90 pruebas de simulación.
- Prueba parte 2 del Algoritmo DMBA-FLPSO:
Utilizando la Metaheurística PSO junto con los resultados de la técnica FL, son 90 pruebas de simulación.
- Prueba para determinar la cantidad de ráfagas generadas, recibidas y perdidas:
Son 90 pruebas de simulación.

Cada una de las pruebas de simulación dependiendo de la carga de tráfico, ver tabla 4.12, se realizan con tiempo de simulación de 0.2, 0.4, 0.6, 0.78 / 0.8 y 0.8 segundos (s), y a una velocidad de datos de 2.5 Gbps. Cabe resaltar que 0.8 segundos de simulación corresponde como mínimo a 6 horas reales, dependiendo de la capacidad del equipo utilizado, por lo que es recomendable utilizar la mayor cantidad de equipos de cómputo simultáneamente para ahorrar tiempo.

Para las pruebas de simulación, con base a la tabla 4.1 y la tabla 4.12, la carga de tráfico se establece según la configuración del Tiempo entre Paquetes (T_p) y el Tiempo de Simulación (T_s), es decir, la relación de tiempos (T_p , T_s), como se muestra en la tabla 4.14.

Tabla 4. 14 Carga de Tráfico según el Tiempo entre Paquetes (T_p) y el Tiempo de Simulación (T_s)

Carga de Tráfico	Tiempo entre Paquetes con Tiempo de Simulación (T_p , T_s)	
	Muy Baja	0.1E
0.2E		(4.9 μ s, 0.2s)
Baja	0.3E	(4.7 μ s, 0.4s)
	0.4E	(4.6 μ s, 0.4s)
Media	0.5E	(4.4 μ s, 0.6s)
	0.6E	(4.3 μ s, 0.6s)
Alta	0.7E	(4.1 μ s , 0.78s/0.78s)
	0.8E	(4.0 μ s, 0.78s/0.78s)
Muy Alta	0.9E	(3.8 μ s, 0.8s/0.8s /1.2s)
	1E	(3.7 μ s, 0.8s/0.8s /1.2s)



4. 6 ANÁLISIS DE RESULTADOS

Se presentan los resultados obtenidos de las Pruebas de Simulación según el valor de los parámetros de ensamble, uL y T , la carga de tráfico y la Semilla inicial generadora de tráfico en una red OBS Distribuida (red de prueba) sobre la herramienta de simulación OMNeT++. Además, se analiza los resultados utilizando los parámetros de desempeño, la Probabilidad de Bloqueo de Ráfagas ($PB_{ráfagas}$) y el Retardo extremo a extremo de transmisión de Ráfagas ($e2e-D_{ráfagas}$), donde, los datos-resultados son exportados hacia la herramienta ORIGIN PRO 8.6, para la representación gráfica y análisis de los datos.

Tanto para la configuración que implementa el algoritmo de ensamble mixto Clásico como para el algoritmo DMBAА-FLPSO, se varía el tráfico en la red, el cual ha sido normalizado (valores de 0.1 a 1 Erlang, con intervalos de 0.1). A continuación, se presentan algunas de las pruebas de simulación.

4.6 1 Pruebas de Simulación para Carga Muy Baja, sin/con el Algoritmo DMBAА-FLPSO, con Semilla 108, Semilla 9, para $uL=(2, 3, 4)$ y $T=(2.75, 2.5, 2.25)$. Determinando la $PB_{ráfagas}$ y el $e2e-D_{ráfagas}$

1) Para Carga Muy Baja con 0.1E (5.0 μ s, 0.2s/0.2s) y Semilla 108:

Las figuras 4.10, 4.11 y 4.12, muestran la probabilidad de bloqueo de ráfagas ($PB_{ráfagas}$) por nodo en la red de prueba (OBS-Distribuida), para carga de tráfico Muy Baja con 0.1E (5.0 μ s, 0.2s/0.2s), Semilla 108 y la combinación de los valores establecidos para uL y T

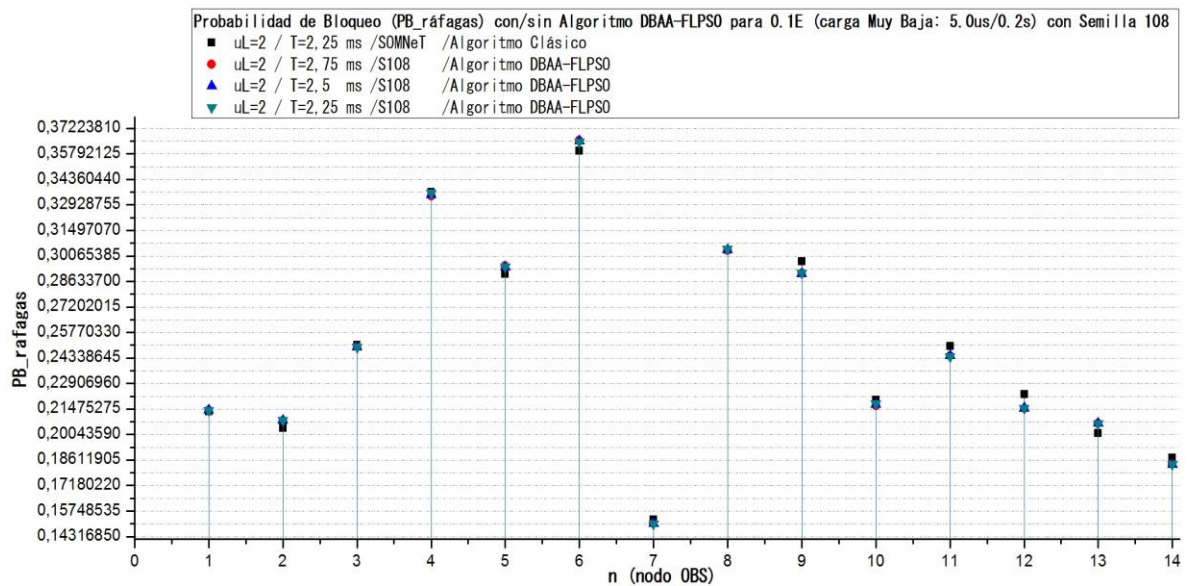


Figura 4. 10 PB por nodo para Carga Muy Baja con 0.1E, con $uL=2/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAА-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

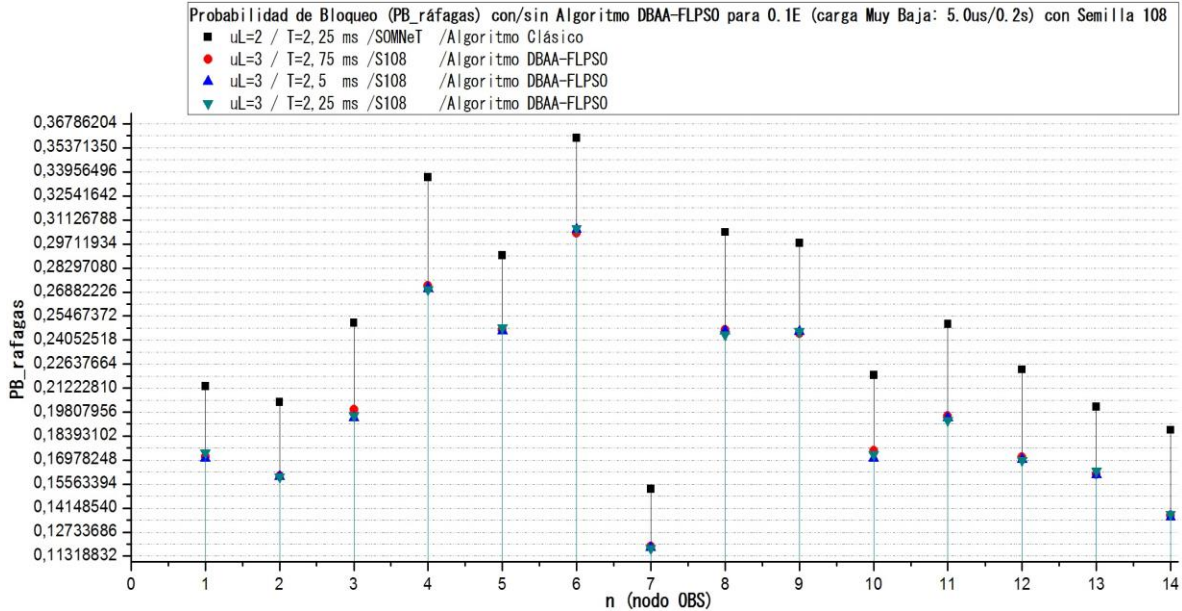


Figura 4. 11 PB por nodo para Carga Muy Baja con 0.1E, con uL=3/T=(2.75,2.25,2.5) para el Algoritmo DMBAА-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

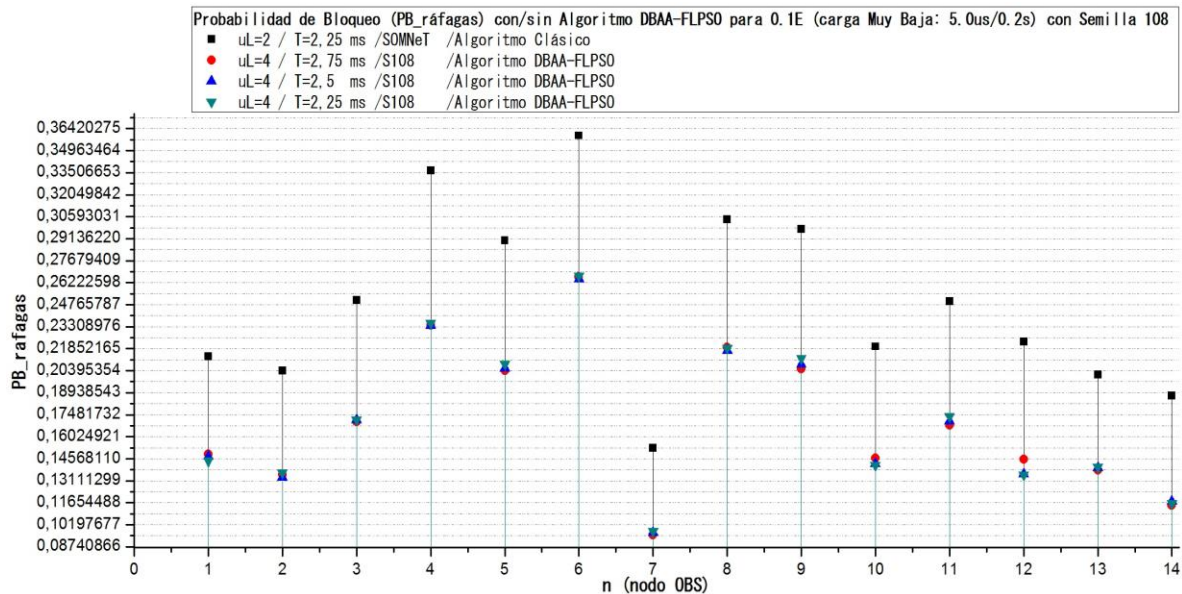


Figura 4. 12 PB por nodo para Carga Muy Baja con 0.1E, con uL=4/T=(2.75,2.25,2.5) para el Algoritmo DMBAА-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

Analizando las figuras 4.10, 4.11 y 4.12, la figura 4.10, se presenta algunas combinaciones uL-T donde la PB_ráfagas es mayor al implementar el algoritmo de ensamble Clásico que con el algoritmo DMBAА-FLPSO. En cambio, en las figuras 4.11 y 4.12, para todas las combinaciones uL-T la PB_ráfagas es menor (mejor) al implementar el algoritmo DMBAА-FLPSO. Además, se observa que al aumentar el valor de uL mejora la PB_ráfagas en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.13 y la 4.14, muestran el nodo con mayor PB_ráfagas



(nodo 6) y el nodo con menor PB_ráfagas (nodo 7), respectivamente, presentando a mejor escala la PB_ráfagas para cada combinación uL y T.

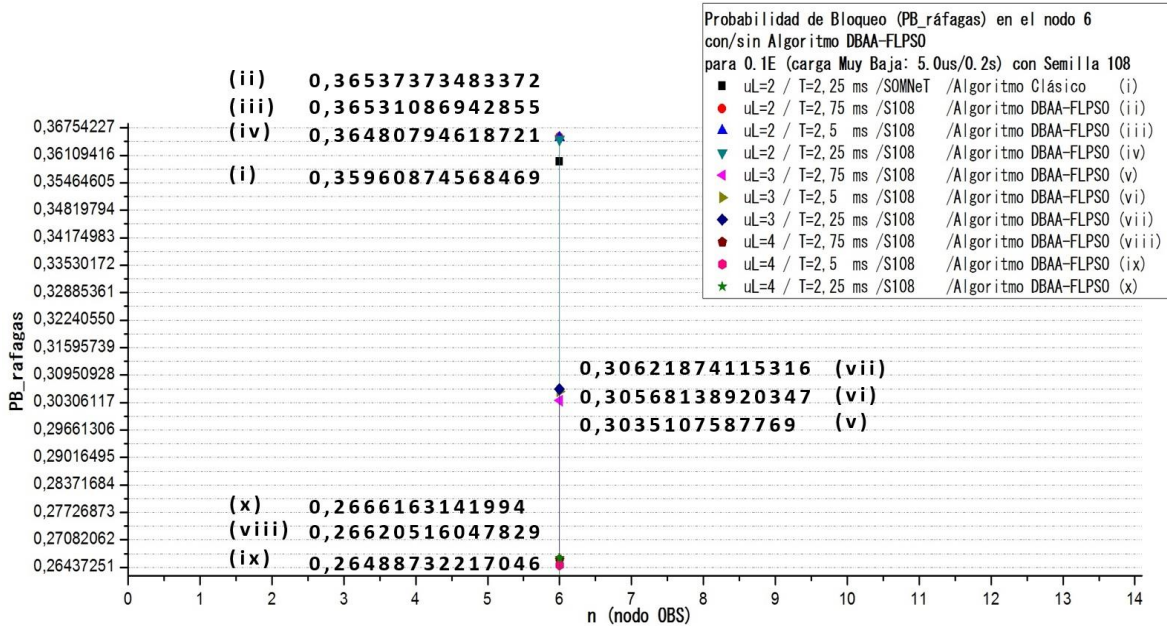


Figura 4. 13 PB nodo 6 para Carga Muy Baja con 0.1E, con uL=(2, 3 4)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

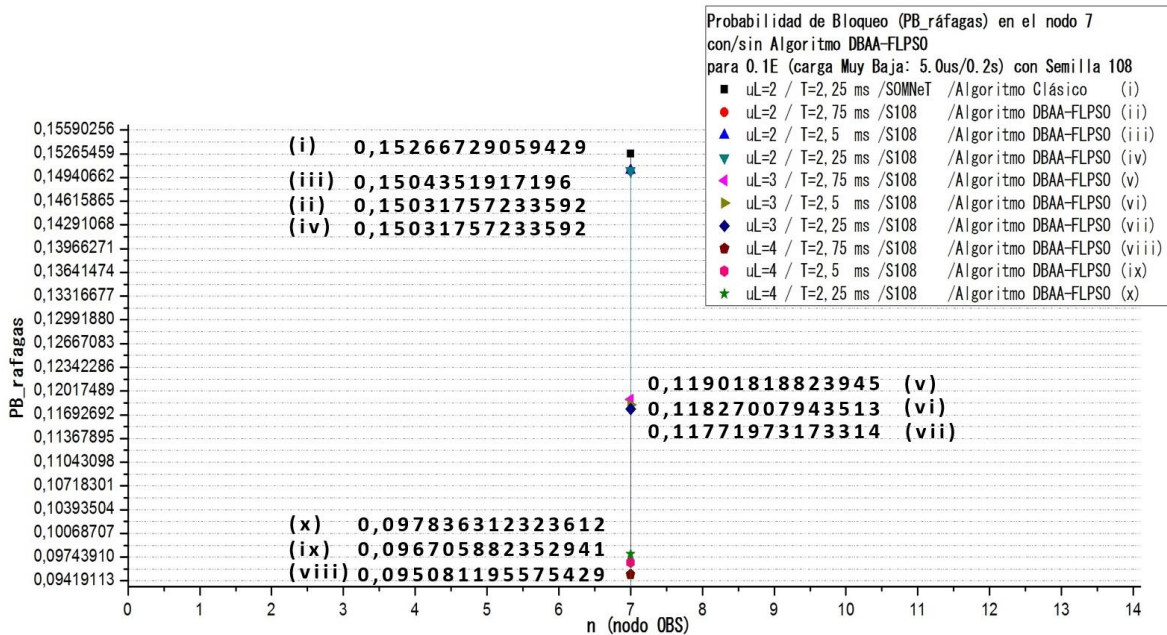


Figura 4. 14 PB nodo 7 para Carga Muy Baja con 0.1E, con uL=(2, 3 4)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

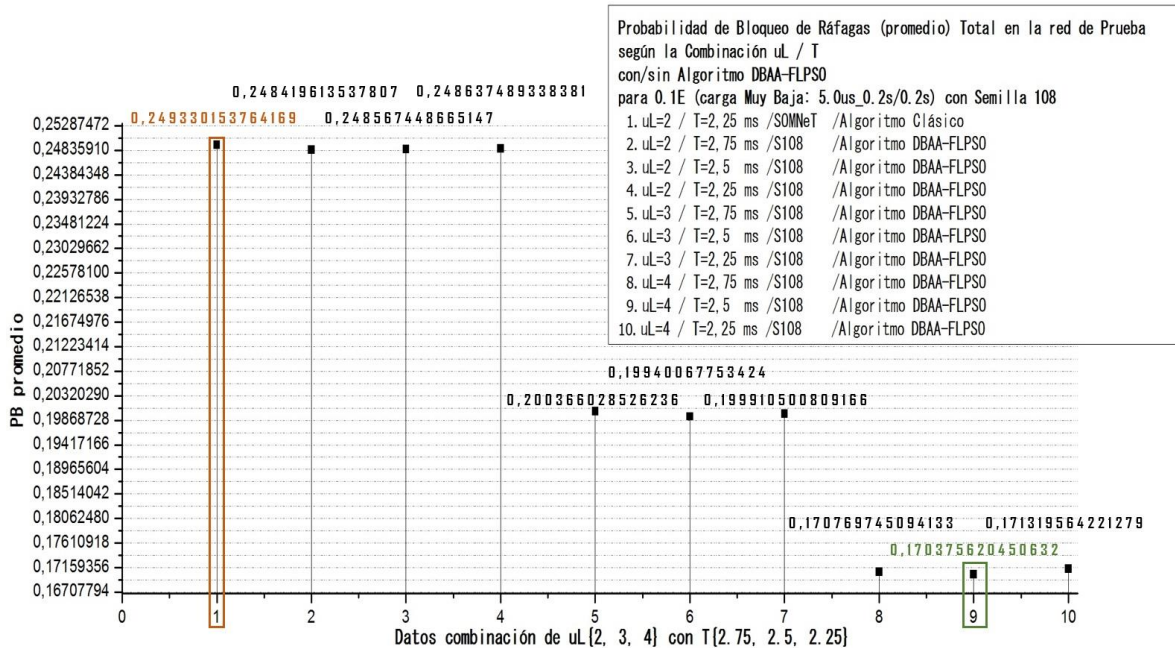


Figura 4. 15 PB promedio por red para Carga Muy Baja con 0.1E, con uL=(2, 3, 4)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Muy Baja con 0.1E, ver figura 4.15, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.15, la **PB_ráfagas= 0,249330153764169** con **uL=2 y T=2.25ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.15) y la **PB_ráfagas= 0,170375620450632** con **uL=4 y T=2.5ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.15).

Tabla 4. 15 PB_ráfagas para Carga Muy Baja 0.1E con y sin el algoritmo DMBAA-FLPSO

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Muy Baja (5.0µs, 0.2s/0.2s)	0.1E	PB_ráfagas= 0,249330153764169 con uL=2 y T=2.25ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,170375620450632 con uL=4 y T=2.5ms Algoritmo DMBAA-FLPSO



2) Carga Muy Baja para 0.2E (4.9µs, 0.2s/0.2s) con Semilla 9:

Las figuras 4.16, 4.17 y 4.18, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Muy Baja con 0.2E (4.9µs, 0.2s/0.2s), Semilla 9 y la combinación de los valores establecidos para uL y T.

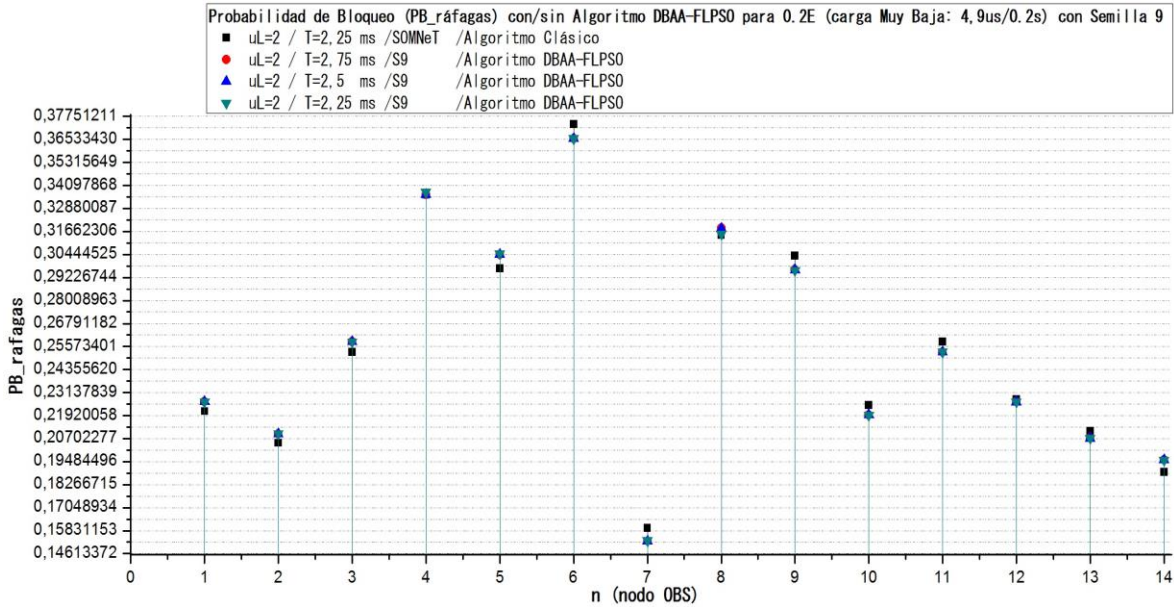


Figura 4. 16 PB por nodo para Carga Muy Baja con 0.2E, con uL=2/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

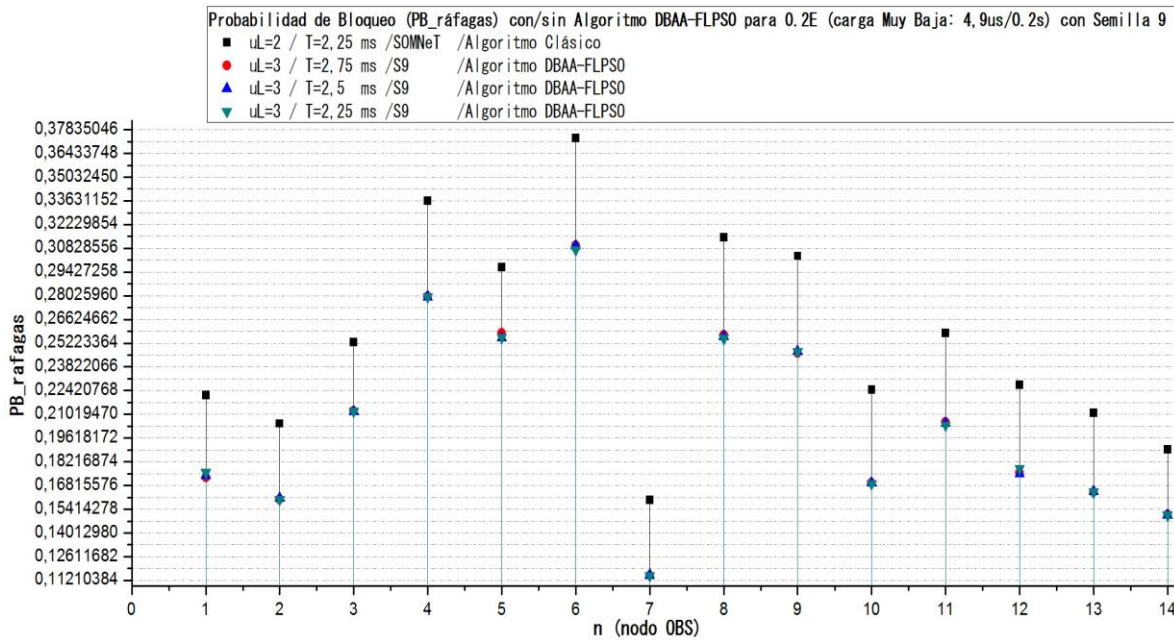


Figura 4. 17 PB por nodo para Carga Muy Baja con 0.2E, con uL=3/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

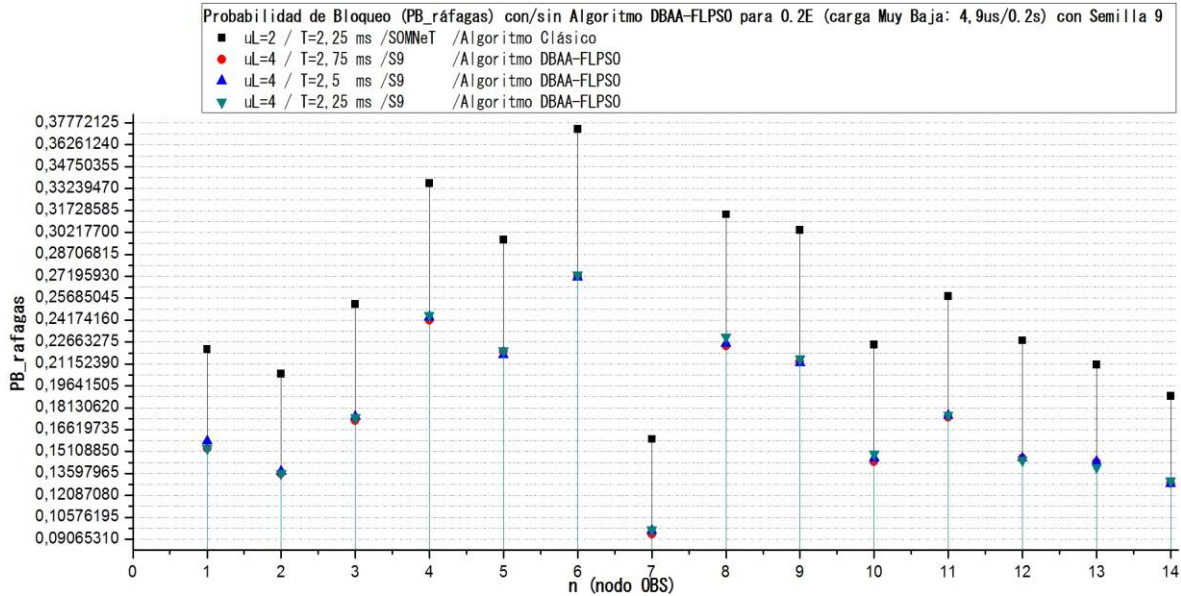


Figura 4. 18 PB por nodo para Carga Muy Baja con 0.4E, con $uL=2/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.

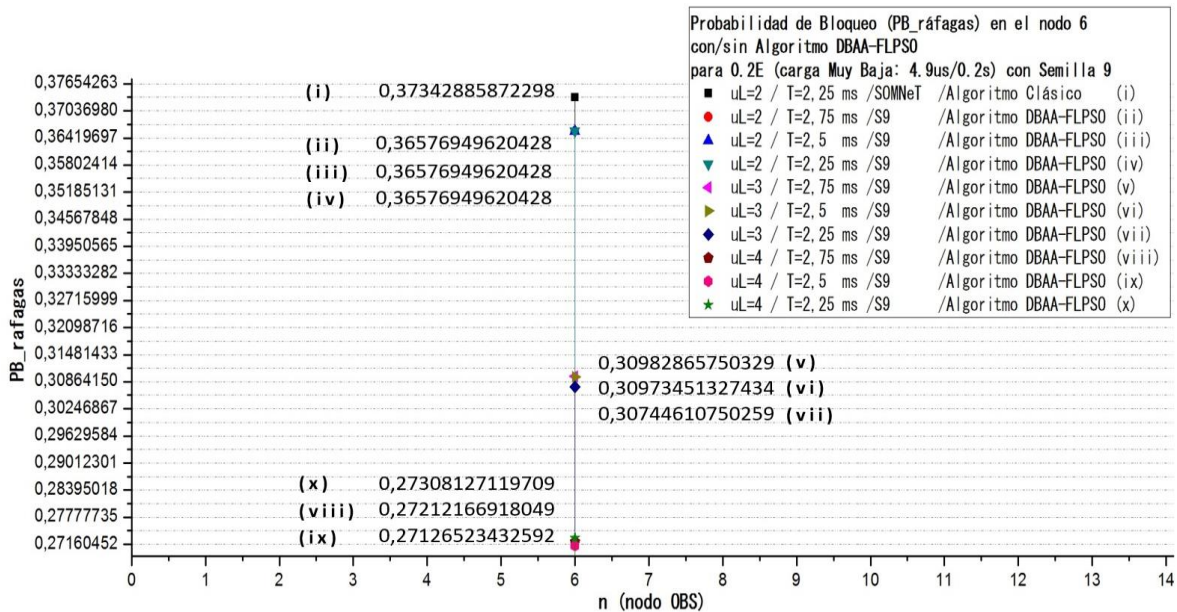


Figura 4. 19 PB nodo 6 para Carga Muy Baja con 0.2E, con $uL=(2, 3 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.

Analizando las figuras 4.16, 4.17 y 4.18, la figura 4.16, se presenta algunas combinaciones $uL-T$ donde la $PB_{ráfagas}$ es mayor al implementar el algoritmo de ensamble Clásico que con el algoritmo DMBAA-FLPSO. En cambio, en las figuras 4.17 y 4.18, para todas las combinaciones $uL-T$ la $PB_{ráfagas}$ es menor (mejor) al implementar el algoritmo DMBAA-FLPSO. Además, se observa que al aumentar el valor de uL mejora la $PB_{ráfagas}$ en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.19 y la 4.20, muestran el nodo con mayor $PB_{ráfagas}$



(nodo 6) y el nodo con menor PB_ráfagas (nodo 7), respectivamente, presentando a mejor escala la PB_ráfagas para cada combinación uL y T.

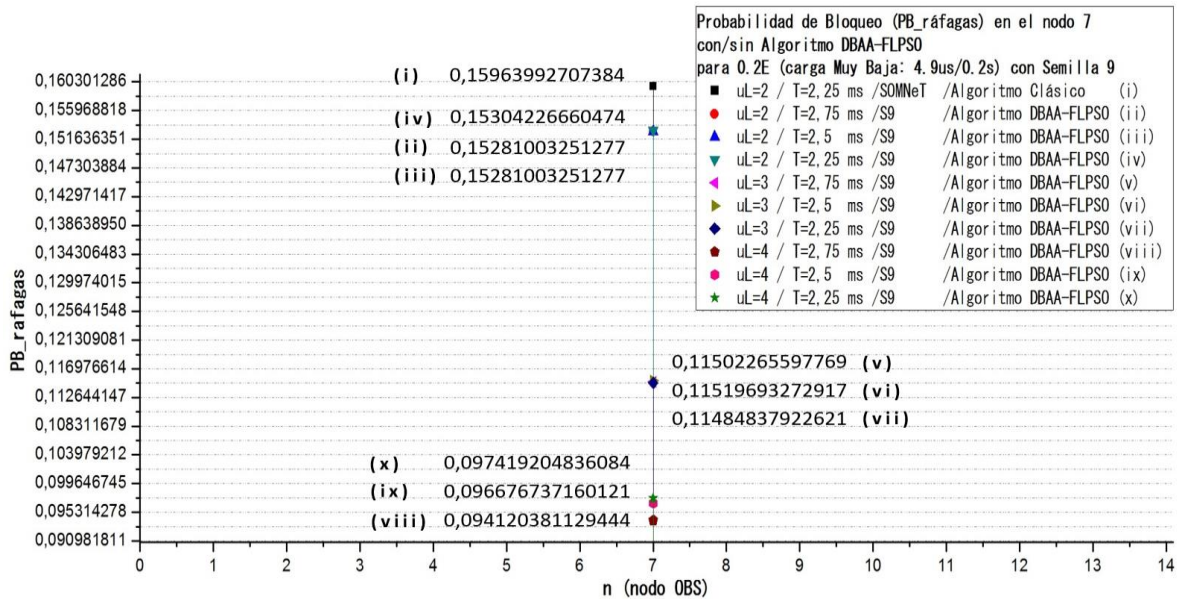


Figura 4. 20 PB nodo 7 para Carga Muy Baja con 0.2E, con uL=(2, 3 4)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

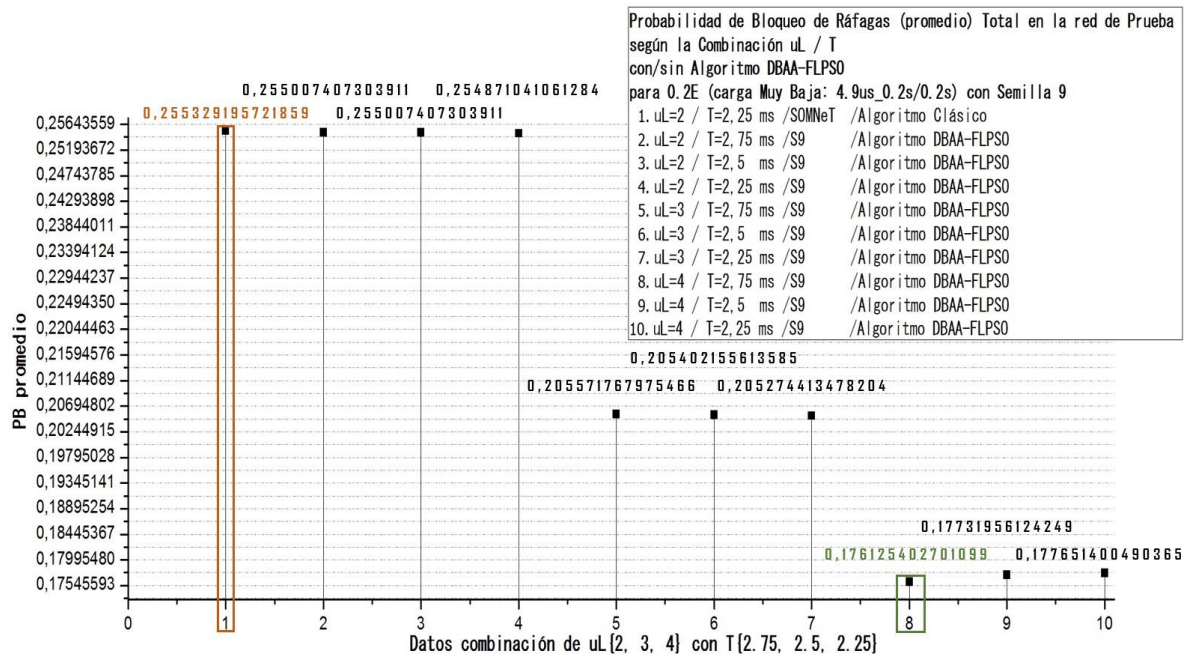


Figura 4. 21 PB promedio por red para Carga Muy Baja con 0.2E, con uL=(2, 3, 4)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Muy Baja con 0.2E, ver figura 4.21, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor



valor (el peor de los casos) y al implementar el algoritmo DMBAА-FLPSO. Donde, al implementar el algoritmo DMBAА-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.16, la **PB_ráfagas= 0,255329195721859** con **uL=2 y T=2.25ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.21) y la **PB_ráfagas= 0,176125402701099** con **uL=4 y T=2.75ms** al implementar el algoritmo DMBAА-FLPSO (en color verde, ver figura 4.21).

Tabla 4. 16 PB_ráfagas para Carga Muy Baja 0.2E con y sin el algoritmo DMBAА-FLPSO

Carga de Tráfico		PB_ráfagas con uL y T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAА-FLPSO)
Muy Baja (4.9µs, 0.2s/0.2s)	0.2E	PB_ráfagas= 0,255329195721859 con uL=2 y T=2.25ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,176125402701099 con uL=4 y T=2.75ms Algoritmo DMBAА-FLPSO

4.6 2 Pruebas de Simulación para Carga Baja, sin/con el Algoritmo DMBAА-FLPSO, con Semilla 40187 /Semilla 7385 y para uL=(2, 3, 4) y T=(2.75, 2.5, 2.25). Determinando PB_ráfagas y e2e-D_ráfagas

3) Carga Baja para 0.3E (4.7µs, 0.4s/0.4s) con Semilla 40187:

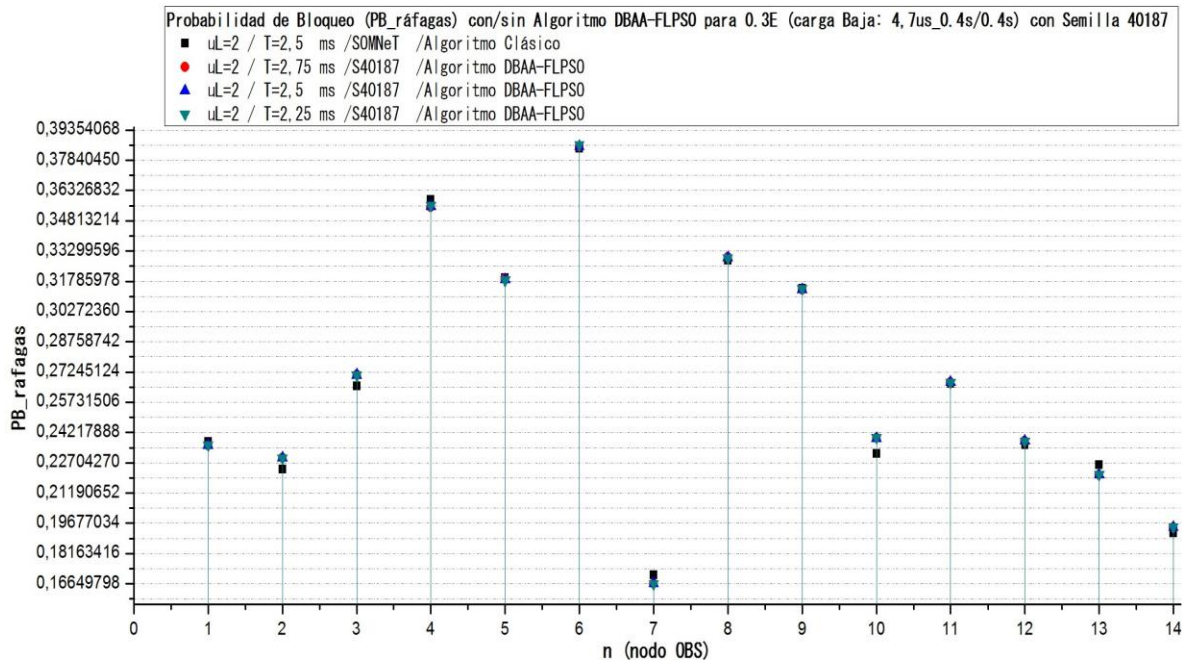


Figura 4. 22 PB por nodo para Carga Baja con 0.3E, con uL=2/T=(2.75,2.25,2.5) para el Algoritmo DMBAА-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

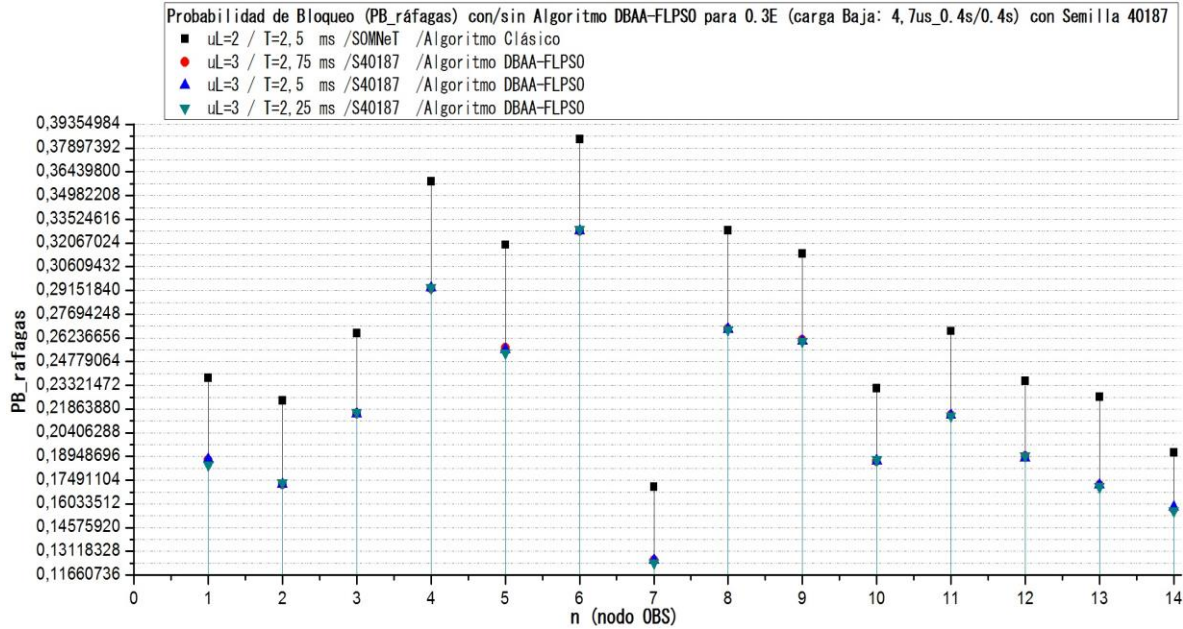


Figura 4. 23 PB por nodo para Carga Baja con 0.3E, con $uL=3/T=(2.75,2.25,2.5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

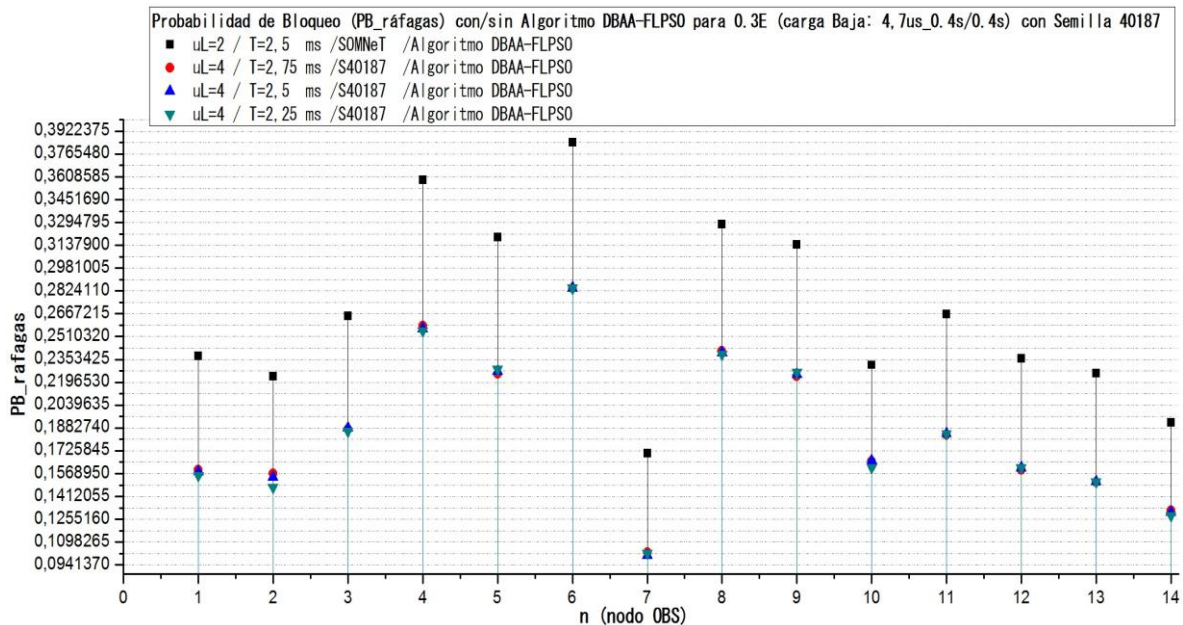


Figura 4. 24 PB por nodo para Carga Baja con 0.3E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

Las figuras 4.22, 4.23 y 4.24, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para una carga de tráfico Baja con 0.3E (4.7μs, 0.4s/0.4s), Semilla 40187 y la combinación de los valores establecidos para uL y T.



Analizando las figuras 4.22, 4.23 y 4.24, la figura 4.22, se presenta algunas combinaciones uL-T donde la PB_ráfagas es mayor al implementar el algoritmo de ensamble Clásico que con el algoritmo DMBAA-FLPSO. En cambio, en las figuras 4.23 y 4.24, para todas las combinaciones uL-T la PB_ráfagas es menor (mejor) al implementar el algoritmo DMBAA-FLPSO. Además, se observa que al aumentar el valor de uL mejora la PB_ráfagas en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.25 y la 4.26, muestran el nodo con mayor PB_ráfagas (nodo 6) y el nodo con menor PB_ráfagas (nodo 7), respectivamente, presentando a mejor escala la PB_ráfagas para cada combinación uL y T.

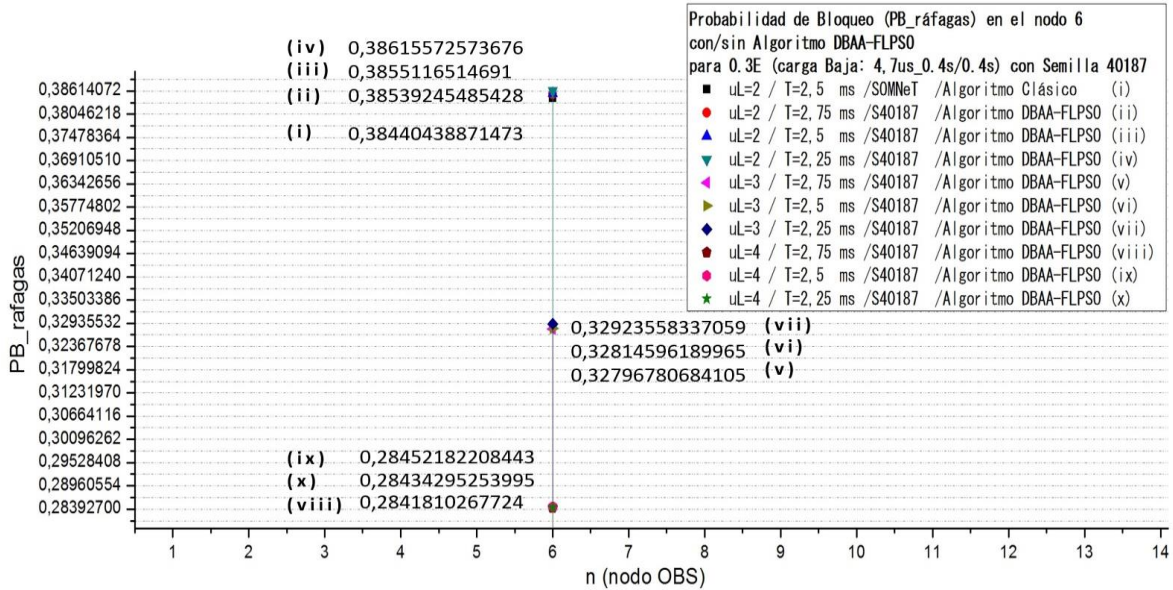


Figura 4. 25 PB nodo 6 para Carga Baja con 0.3E, con uL=(2, 3 4)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

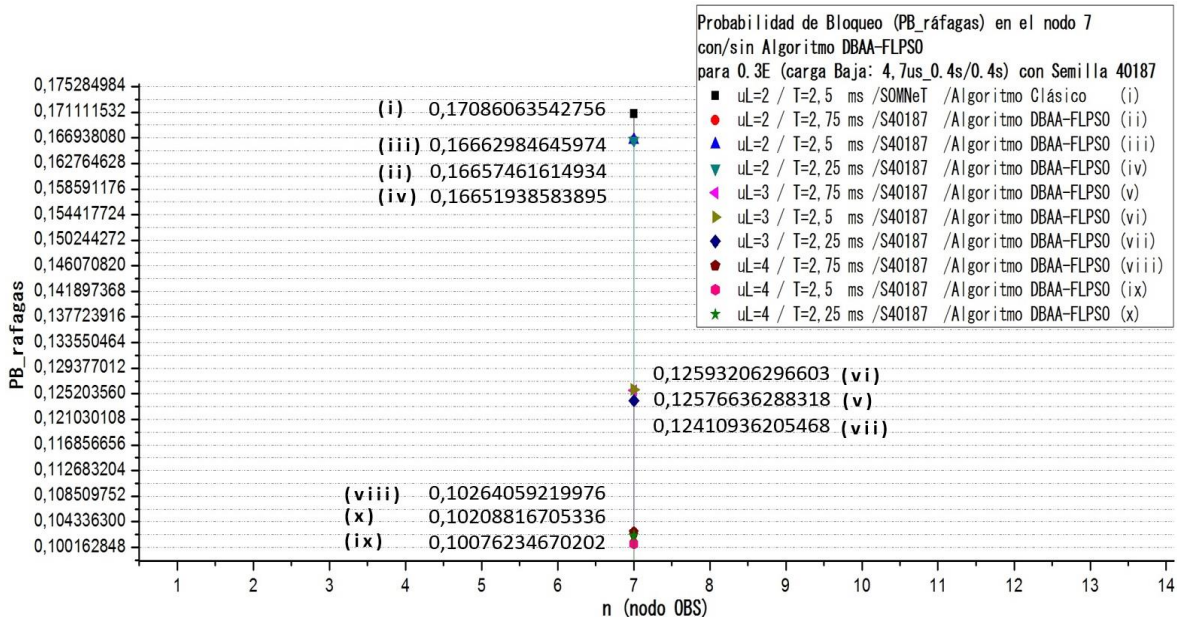


Figura 4. 26 PB nodo 7 para Carga Baja con 0.3E, con uL=(2, 3 4)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

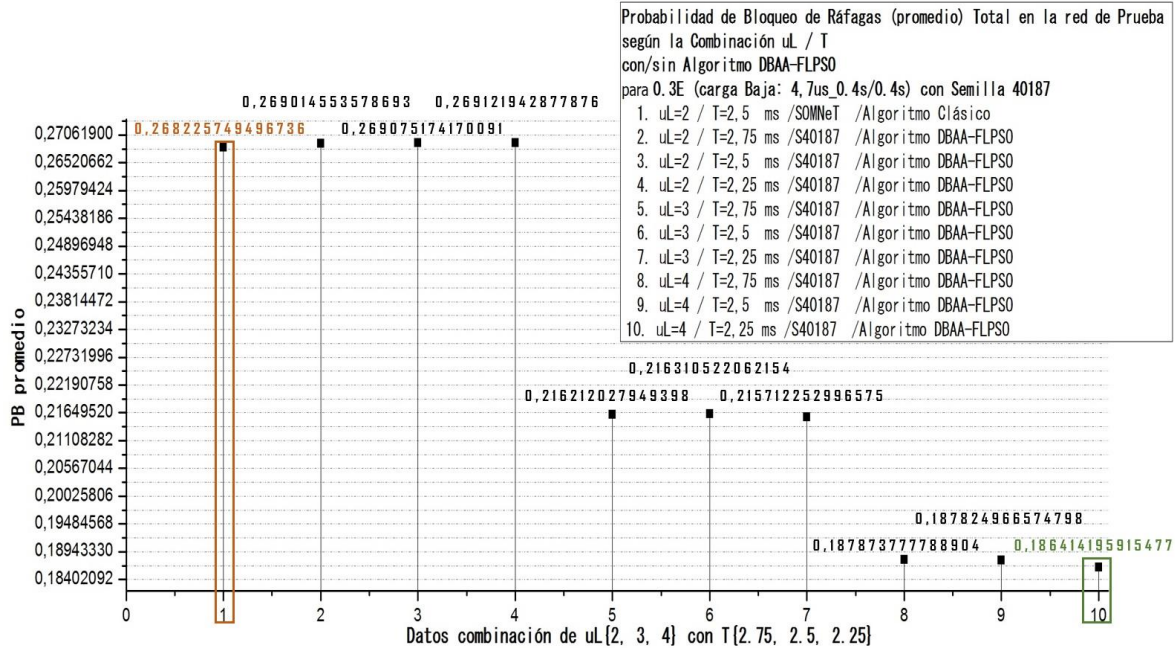


Figura 4. 27 PB promedio por red para Carga Baja con 0.3E, con uL=(2, 3, 4)/T=(2.75, 2.5, 2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Baja con 0.3E, ver figura 4.27, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.17, la **PB_ráfagas= 0,268225749496736** con **uL=2 y T=2.5ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.27) y la **PB_ráfagas= 0,186414195915477** con **uL=4 y T=2.25ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.27).

Tabla 4. 17 PB_ráfagas para Carga Baja 0.3E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Baja (4.7µs, 0.4s/0.4s)	0.3E	PB_ráfagas= 0,268225749496736 con uL=2 y T=2.5ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,186414195915477 con uL=4 y T=2.25ms. Algoritmo DMBAA-FLPSO



4) Carga Baja para 0.4E (4.6 μ s, 0.4s/0.4s) con Semilla 7385

Las figuras 4.28, 4.29 y 4.30, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Baja con 0.4E (4.6 μ s, 0.4s/0.4s), Semilla 7385 y la combinación de los valores establecidos para uL y T.

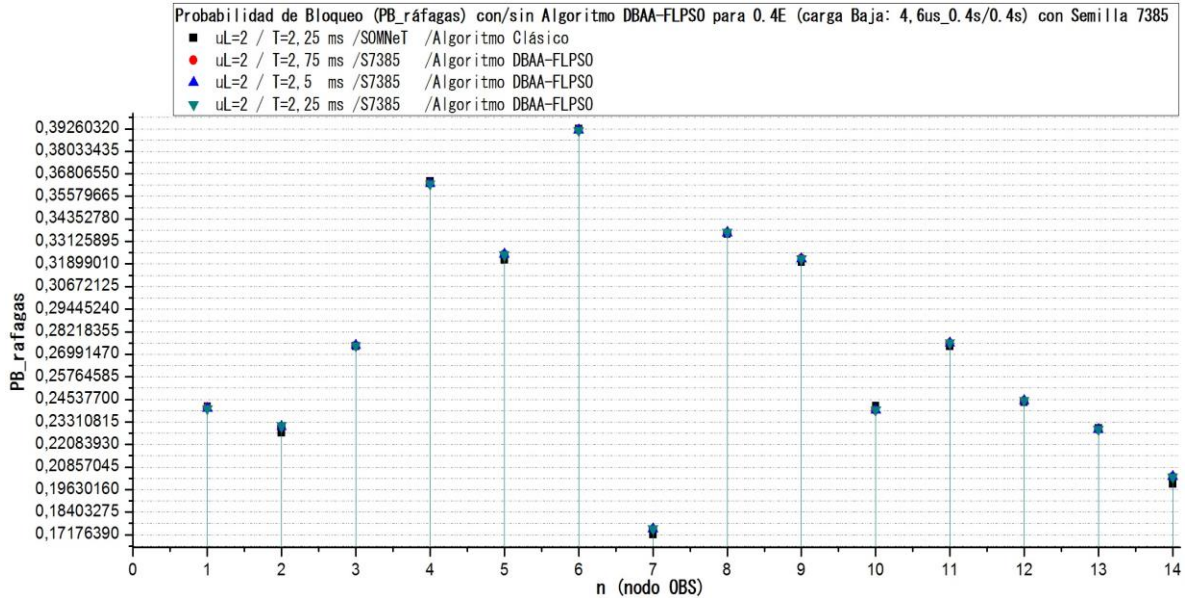


Figura 4. 28 PB por nodo para Carga Baja con 0.4E, con uL=2/T=(2.75,2.25,2.5) para el Algoritmo DMBA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

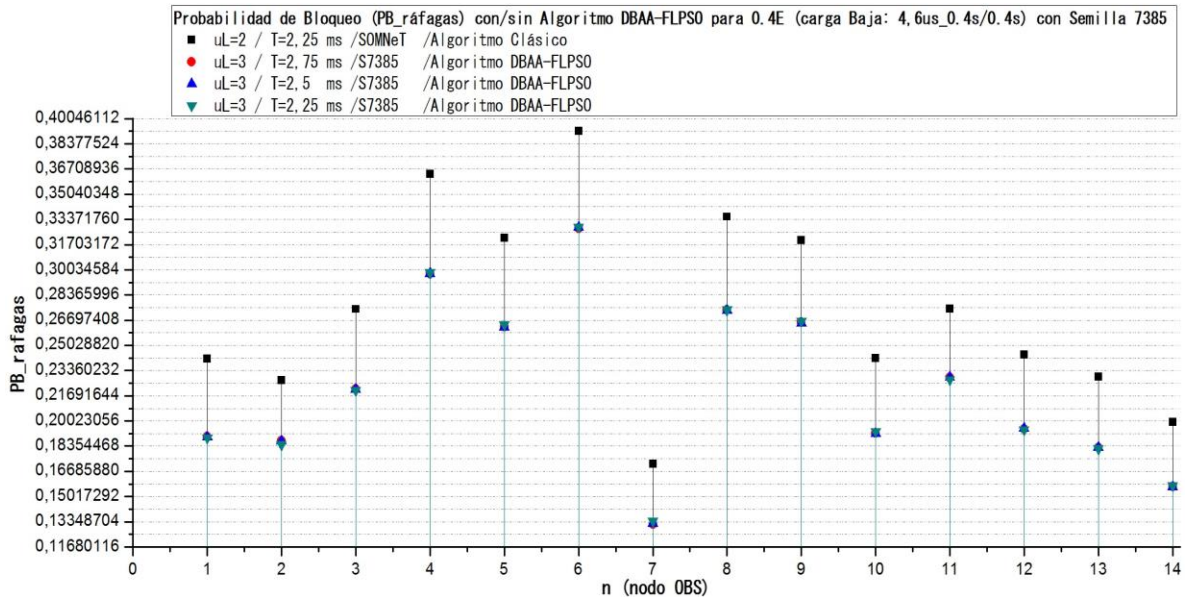


Figura 4. 29 PB por nodo para Carga Baja con 0.4E, con uL=3/T=(2.75,2.25,2.5) para el Algoritmo DMBA-FLPSO y con uL=3/T=2.25 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

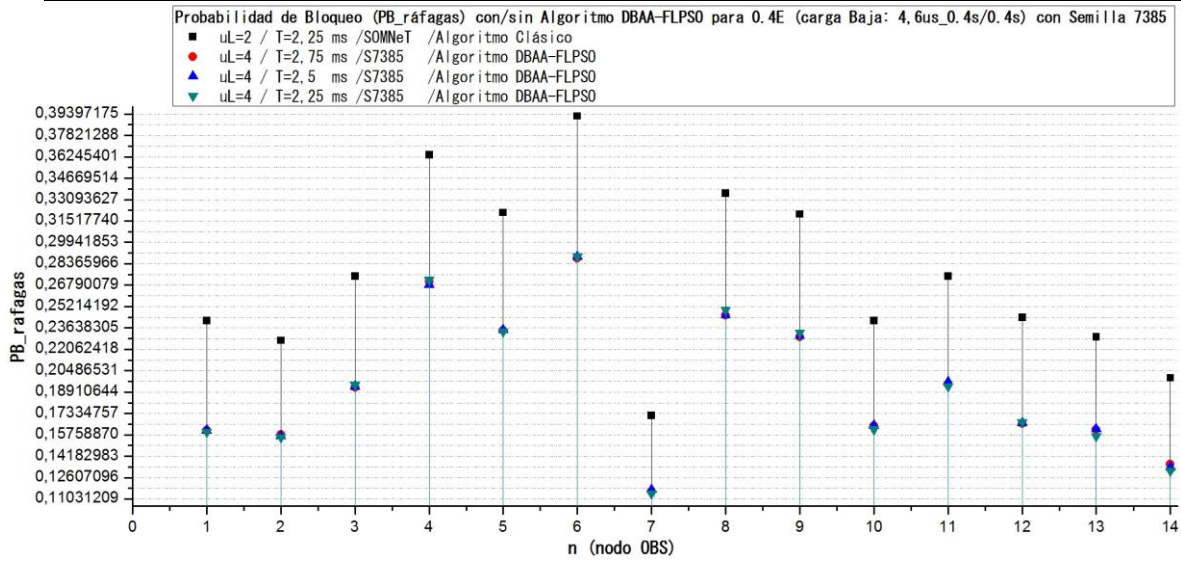


Figura 4. 30 PB por nodo para Carga Baja con 0.4E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.

Analizando las figuras 4.28, 4.29 y 4.30, la figura 4.28, se presenta algunas combinaciones $uL-T$ donde la $PB_{ráfagas}$ es mayor al implementar el algoritmo de ensamble Clásico que con el algoritmo DMBAA-FLPSO. En cambio, en las figuras 4.29 y 4.30, para todas las combinaciones $uL-T$ la $PB_{ráfagas}$ es menor (mejor) al implementar el algoritmo DMBAA-FLPSO. Además, se observa que al aumentar el valor de uL mejora la $PB_{ráfagas}$ en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.31 y la 4.32, muestran el nodo con mayor $PB_{ráfagas}$ (nodo 6) y el nodo con menor $PB_{ráfagas}$ (nodo 7), respectivamente, presentando a mejor escala la $PB_{ráfagas}$ para cada combinación uL y T .

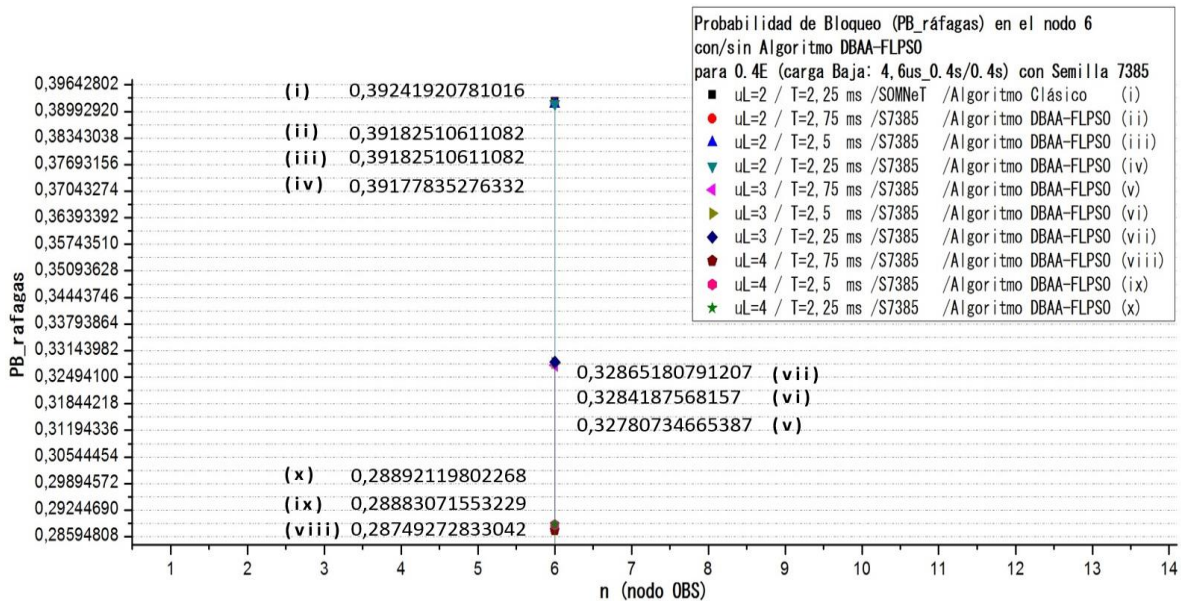


Figura 4. 31 PB nodo 6 para Carga Baja con 0.4E, con $uL=(2, 3 4)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

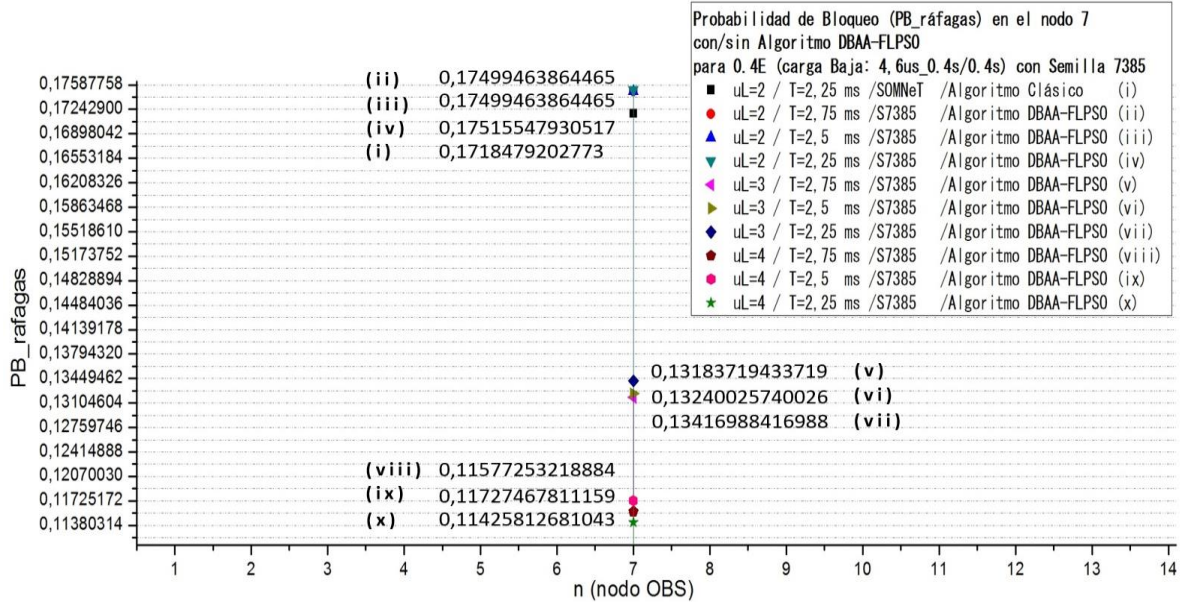


Figura 4. 32 PB nodo 7 para Carga Baja con 0.4E, con uL=(2, 3 4)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

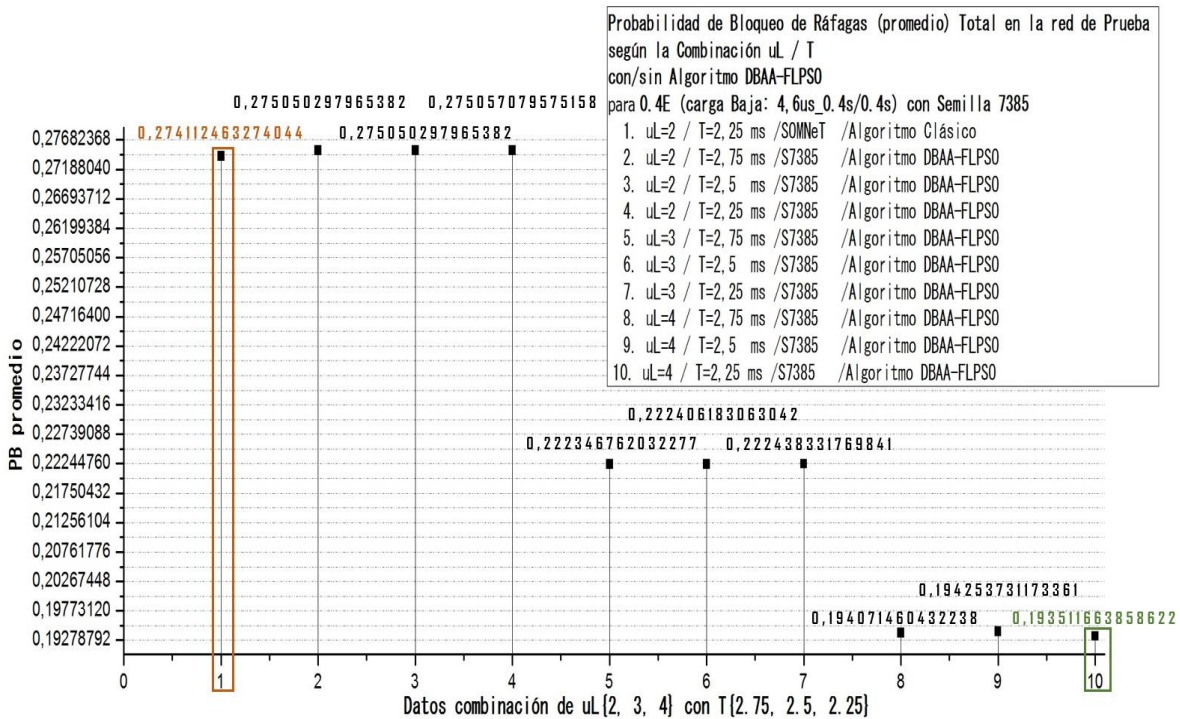


Figura 4. 33 PB promedio por red para Carga Baja con 0.4E, con uL=(2, 3, 4)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Baja con 0.4E, ver figura 4.33, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al



implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.18, la **PB_ráfagas= 0,274112463274044** con **uL=2 y T=2.25ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.33) y la **PB_ráfagas= 0,193511663858622** con **uL=4 y T=2.25ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.33).

Tabla 4. 18 PB_ráfagas para Carga Baja 0.4E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Baja (4.6µs, 0.4s/0.4s)	0.4E	PB_ráfagas= 0,274112463274044 con uL=2 y T=2.25ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,193511663858622 con uL=4 y T=2.25ms. Algoritmo DMBAA-FLPSO

4.6 3 Pruebas de Simulación para Carga Media, sin/con el Algoritmo DMBAA-FLPSO, con Semilla 33 /Semilla 297834 y para uL=(3, 4, 5) y T=(2.75, 2.5, 2.25). Determinando PB_ráfagas y e2e-D_ráfagas

5) Carga Media para 0.5E (4.4µs, 0.6s/0.6s) con Semilla 33:

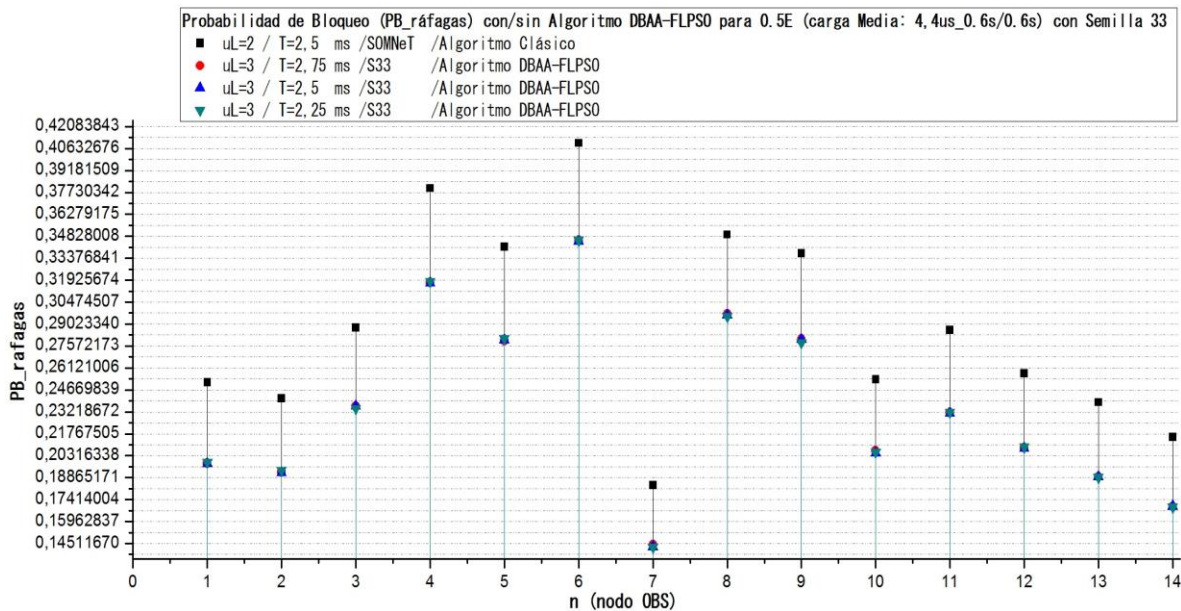


Figura 4. 34 PB por nodo para Carga Media con 0.5E, con uL=3/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

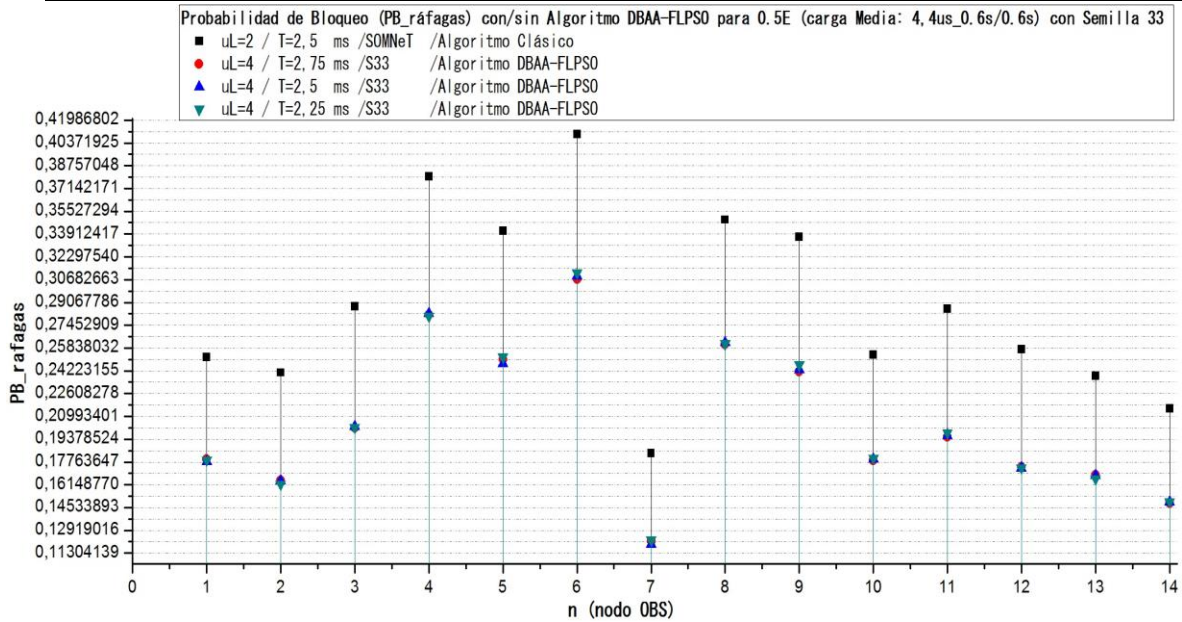


Figura 4. 35 PB por nodo para Carga Media con 0.5E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

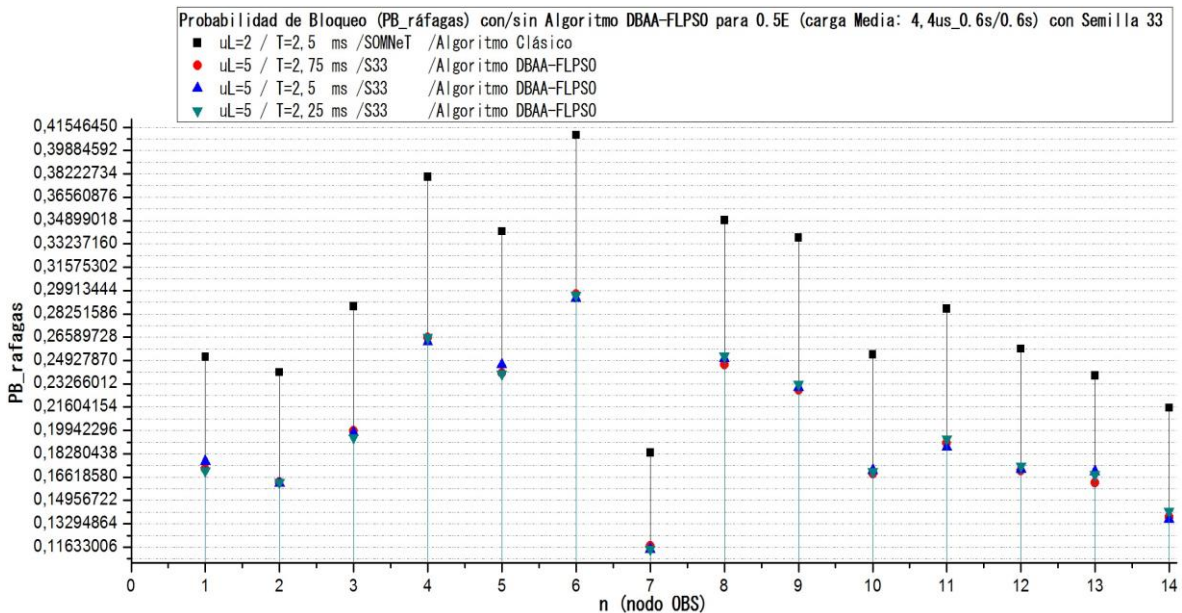


Figura 4. 36 PB por nodo para Carga Media con 0.5E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

Las figuras 4.34, 4.35 y 4.36, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Media con 0.5E (4.4μs, 0.6s/0.6s), Semilla 33 y la combinación de los valores establecidos para uL y T.

Analizando las figuras 4.34, 4.35 y 4.36, en contraste a las figuras para la carga Muy Baja y Carga Baja, con 0.1E, 0.2E, 0.3E y 0.4E respectivamente, para todas



las combinaciones uL-T la PB_ráfagas siempre será menor (mejor) al implementar el algoritmo DBAA-FLPSO que con el algoritmo de ensamble Clásico. Además, se observa que al aumentar el valor de uL mejora la PB_ráfagas en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.37 y la 4.38, muestran el nodo con mayor PB_ráfagas (nodo 6) y el nodo con menor PB_ráfagas (nodo 7), respectivamente, presentando a mejor escala la PB_ráfagas para cada combinación uL y T.

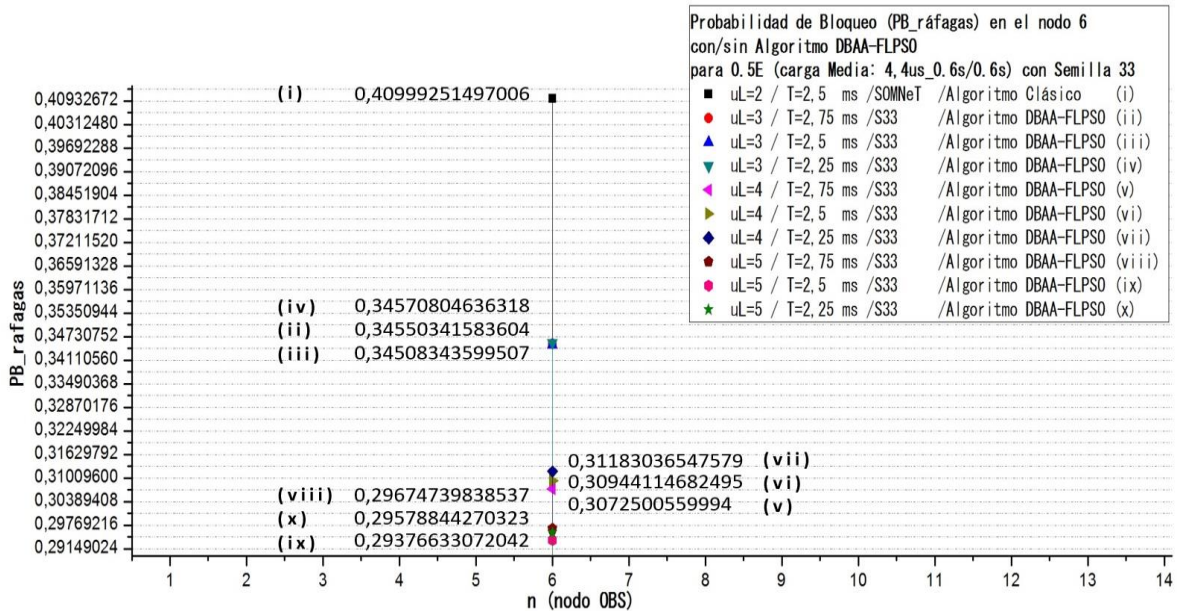


Figura 4. 37 PB nodo 6 para Carga Media con 0.5E, con uL=(3, 4, 5)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

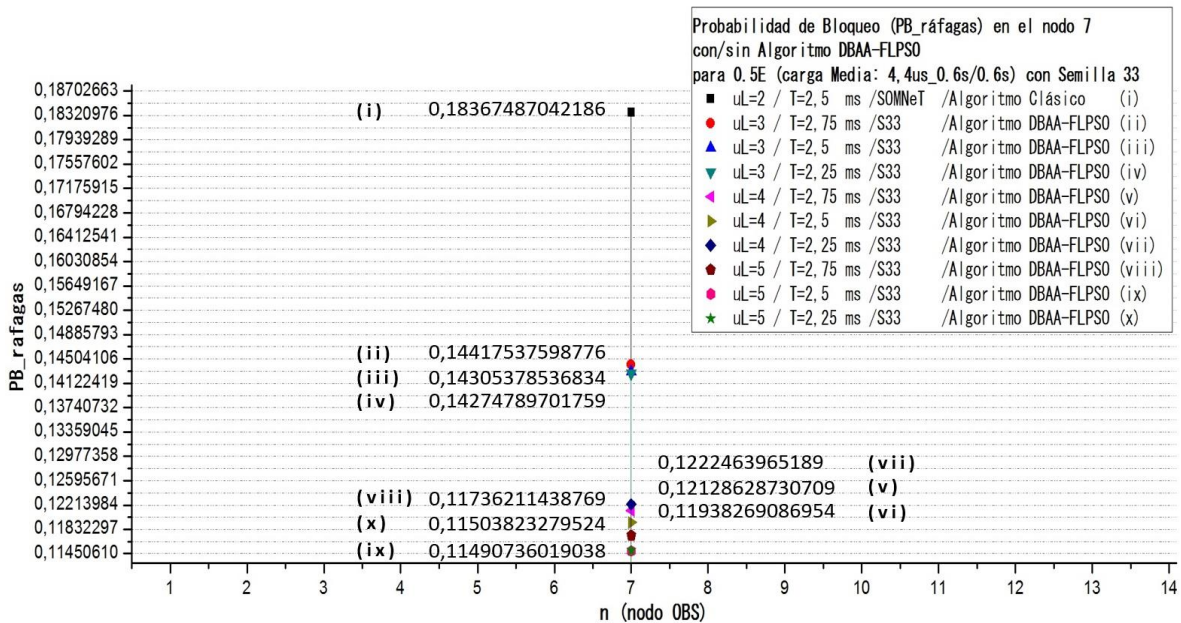


Figura 4. 38 PB nodo 7 para Carga Media con 0.5E, con uL=(3, 4, 5)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

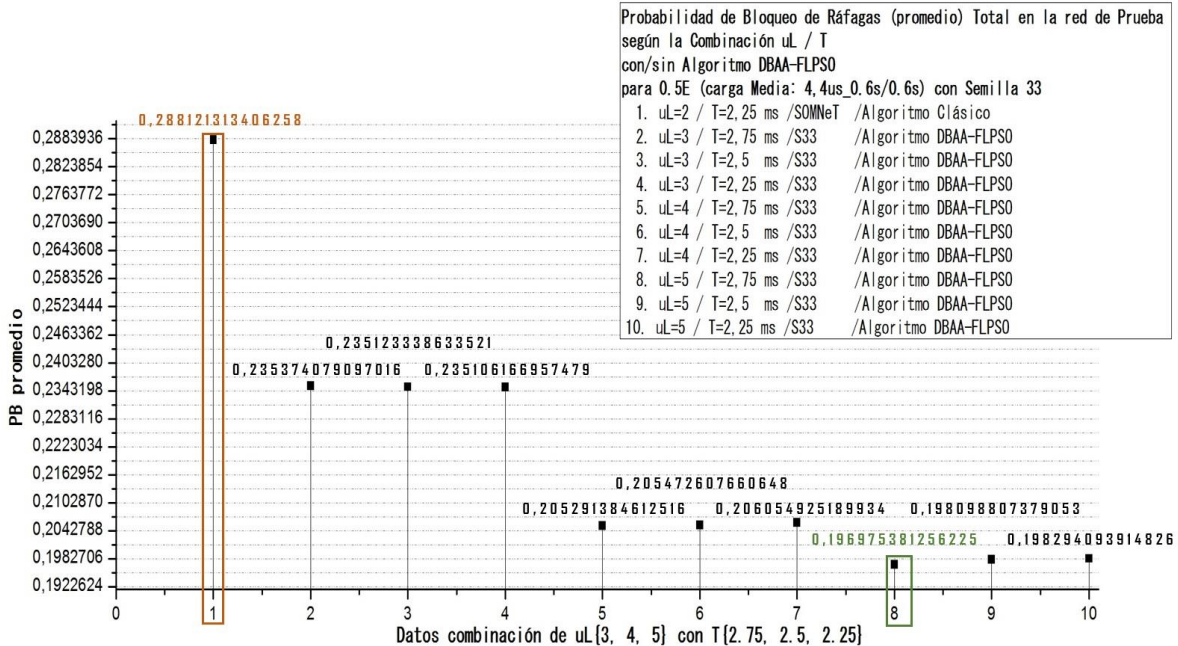


Figura 4. 39 PB promedio por red para Carga Media con 0.5E, con uL=(3, 4, 5)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Media con 0.5E, ver figura 4.39, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.19, la **PB_ráfagas= 0,288121313406258** con **uL=2 y T=2.5ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.39) y la **PB_ráfagas= 0,196975381256225** con **uL=5 y T=2.75ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.39).

Tabla 4. 19 PB_ráfagas para Carga Baja 0.5E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		PB_ráfagas con uL y T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Media (4.4µs, 0.6s/0.6s)	0.5E	PB_ráfagas= 0,288121313406258 con uL=2 y T=2.5ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,196975381256225 con uL=5 y T=2.75ms. Algoritmo DMBAA-FLPSO



6) Carga Media para 0.6E (4.3μs, 0.6s/0.6s) con Semilla 297834:

Las figuras 4.40, 4.41 y 4.42, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Media con 0.6E (4.3μs, 0.6s/0.6s), Semilla 297834 y la combinación de los valores establecidos para uL y T.

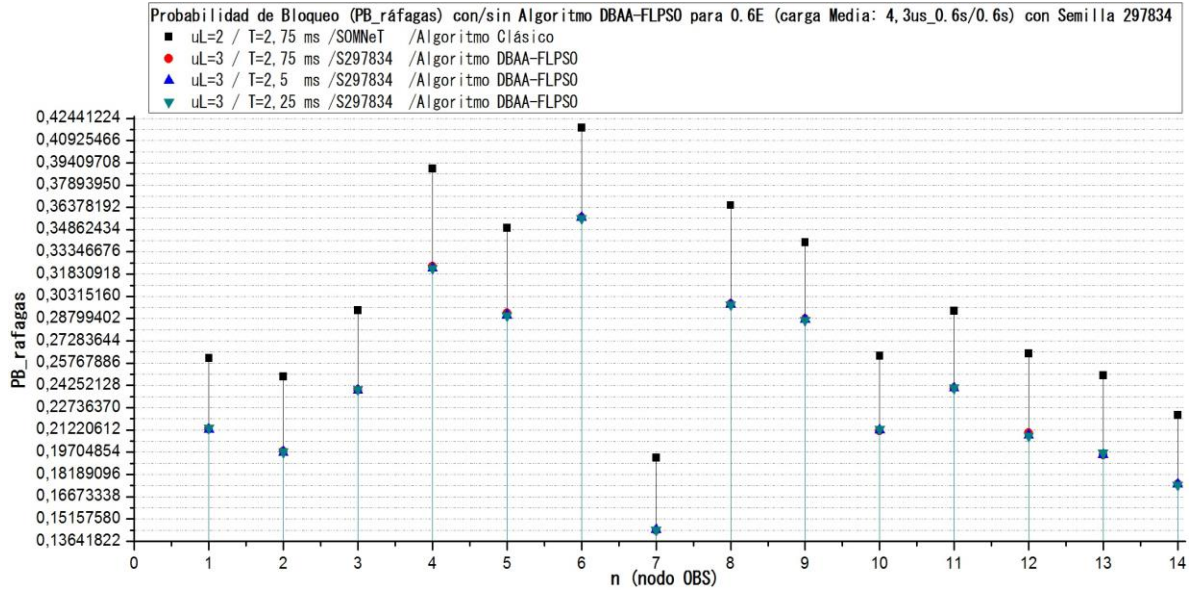


Figura 4. 40 PB por nodo para Carga Media con 0.6E, con uL=3/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.75 para el Algoritmo Clásico.

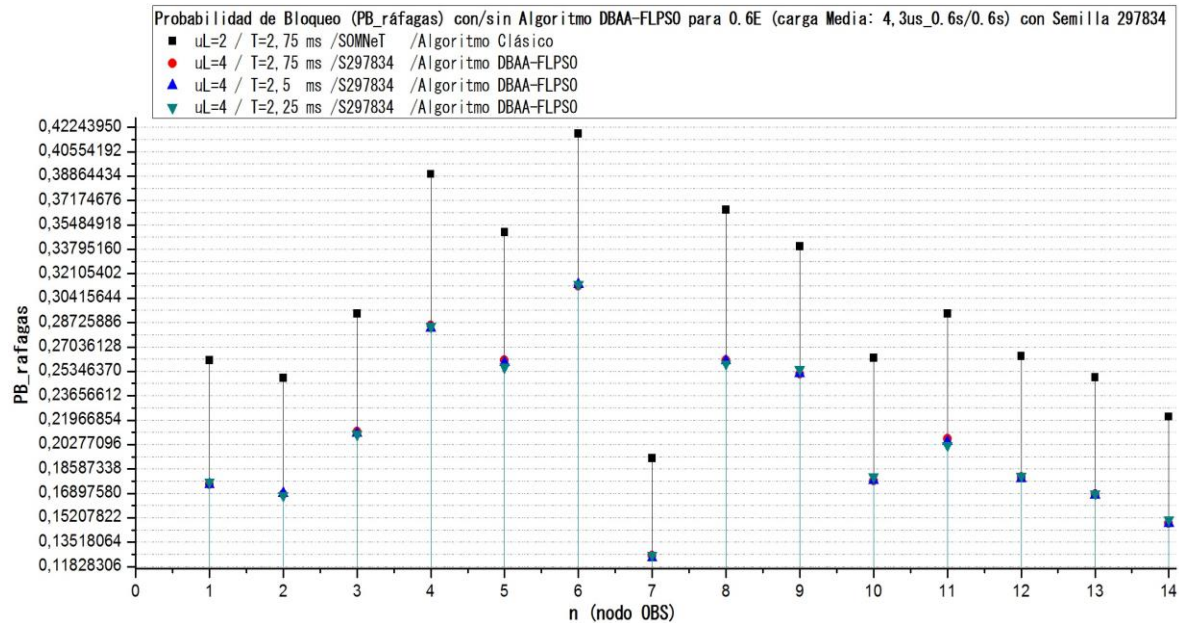


Figura 4. 41 PB por nodo para Carga Media con 0.6E, con uL=4/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.75 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

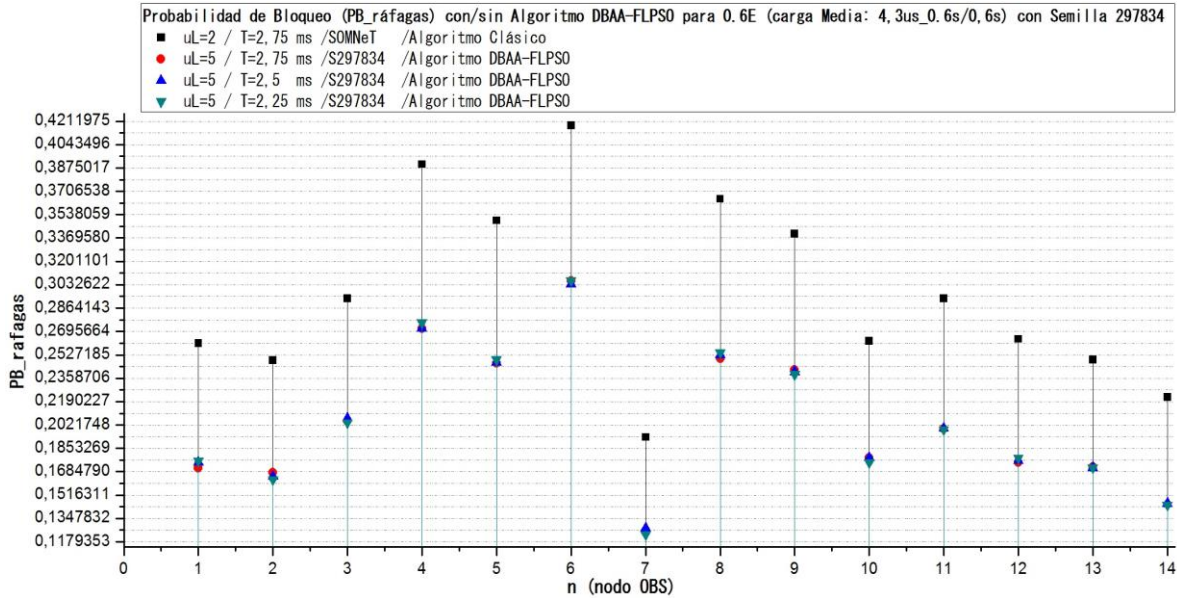


Figura 4. 42 PB por nodo para Carga Media con 0.6E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.

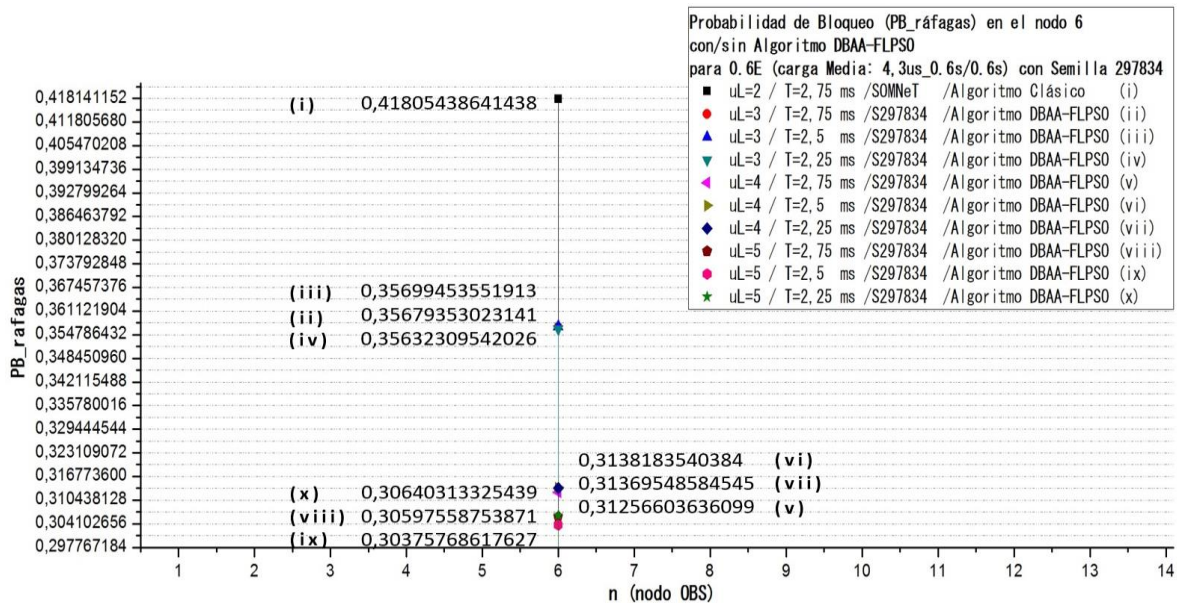


Figura 4. 43 PB nodo 6 para Carga Media con 0.6E, con $uL=(3, 4, 5)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.75$ para el Algoritmo Clásico.

Analizando las figuras 4.40, 4.41 y 4.42, en contraste a las figuras para la carga Muy Baja y Carga Baja, con 0.1E, 0.2E, 0.3E y 0.4E respectivamente, para todas las combinaciones $uL-T$ la $PB_{ráfagas}$ siempre será menor (mejor) al implementar el algoritmo DMBAA-FLPSO que con el algoritmo de ensamble Clásico. Además, se observa que al aumentar el valor de uL mejora la $PB_{ráfagas}$ en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.43 y la 4.44, muestran el nodo con mayor $PB_{ráfagas}$ (nodo 6) y el nodo con menor $PB_{ráfagas}$ (nodo 7),



respectivamente, presentando a mejor escala la PB_ráfagas para cada combinación uL y T.

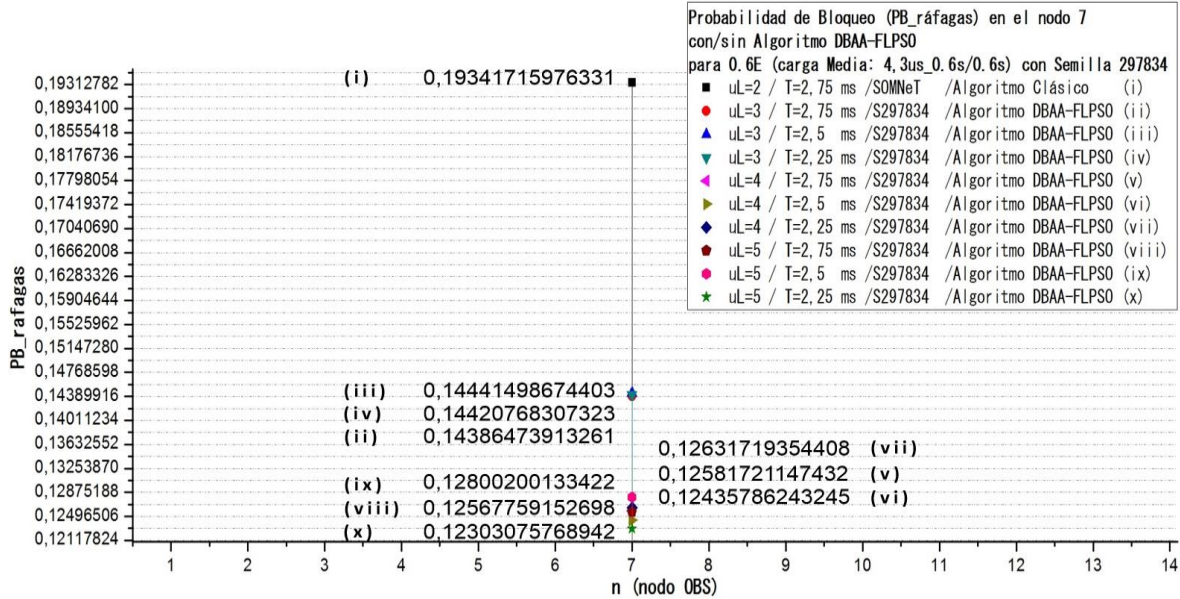


Figura 4. 44 PB nodo 7 para Carga Media con 0.6E, con uL=(3, 4, 5)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.75 para el Algoritmo Clásico.

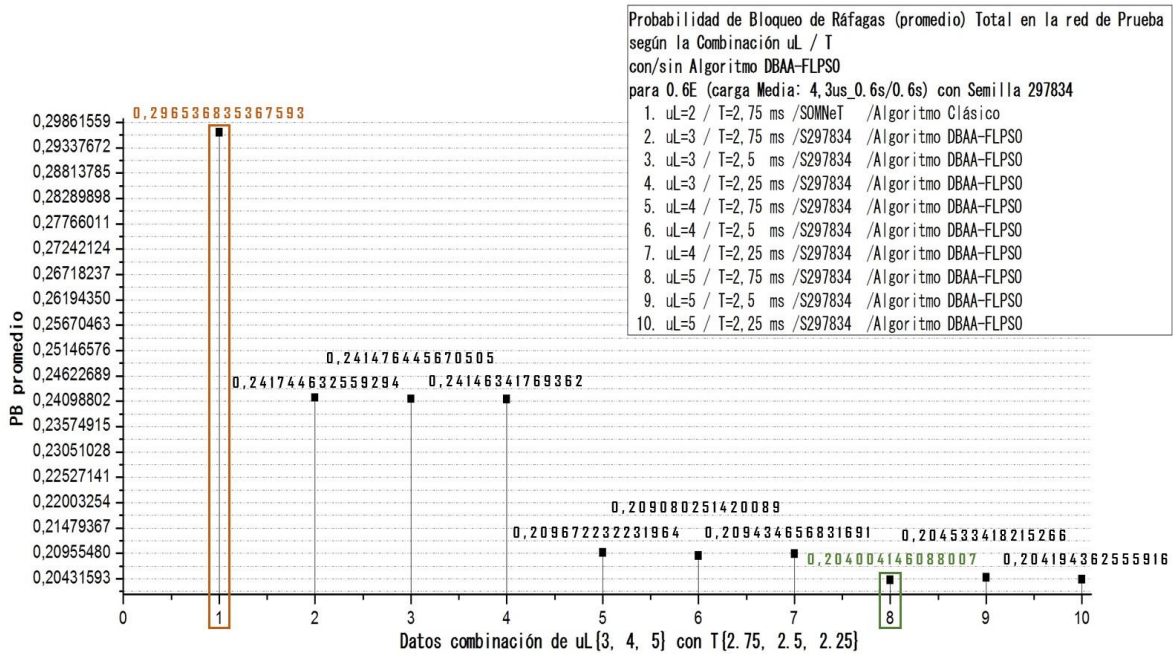


Figura 4. 45 PB promedio por red para Carga Media con 0.6E, con uL=(3, 4, 5)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.75 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Media con 0.6E, ver figura 4.45, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de



ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor $PB_{ráfagas}$ y su combinación de uL y T . De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.20, la $PB_{ráfagas} = 0,296536835367593$ con $uL=2$ y $T=2.75ms$ al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.45) y la $PB_{ráfagas} = 0,204004146088007$ con $uL=5$ y $T=2.75ms$ al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.45).

Tabla 4. 20 $PB_{ráfagas}$ para Carga Baja 0.6E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		$PB_{ráfagas}$ con uL y T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Media (4.3 μ s, 0.6s/0.6s)	0.6E	$PB_{ráfagas} = 0,296536835367593$ con $uL=2$ y $T=2.75ms$. Algoritmo de ensamble Clásico
		$PB_{ráfagas} = 0,204004146088007$ con $uL=5$ y $T=2.75ms$. Algoritmo DMBAA-FLPSO

4.6 4 Pruebas de Simulación para Carga Alta, sin/con el Algoritmo DMBAA-FLPSO, con Semilla 532570 /Semilla 150 y para $uL=(4, 5, 6)$ y $T=(2.75, 2.5, 2.25)$. Determinando $PB_{ráfagas}$ y $e2e-D_{ráfagas}$

7) Carga Alta para 0.7E (4.1 μ s, 0.78s/0.78s) con Semilla 532570:

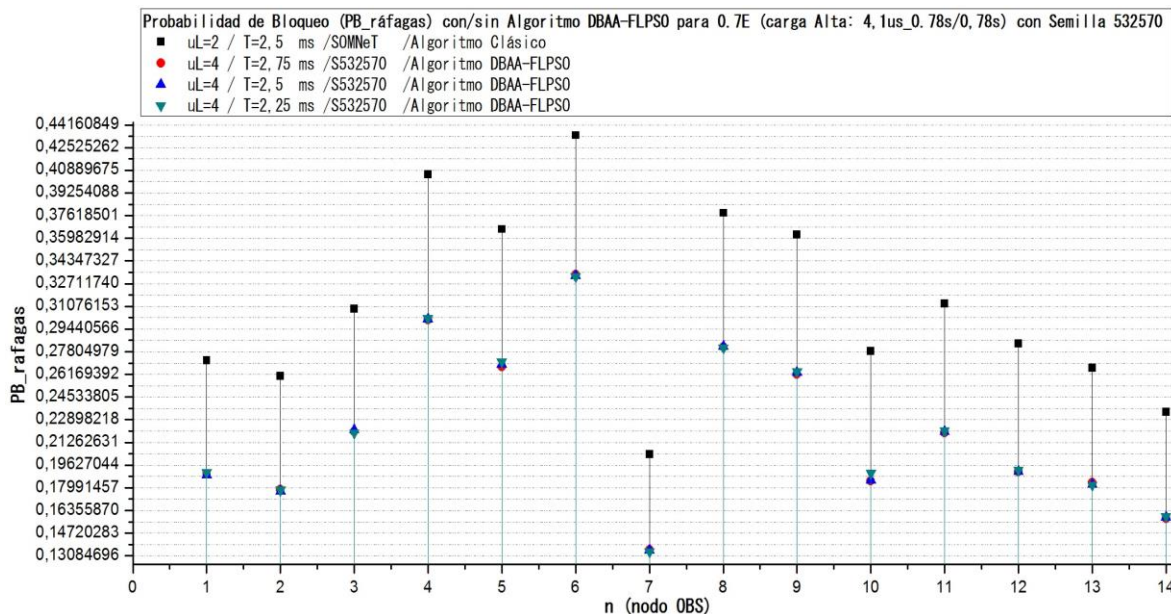


Figura 4. 46 PB por nodo para Carga Alta con 0.7E, con $uL=4/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

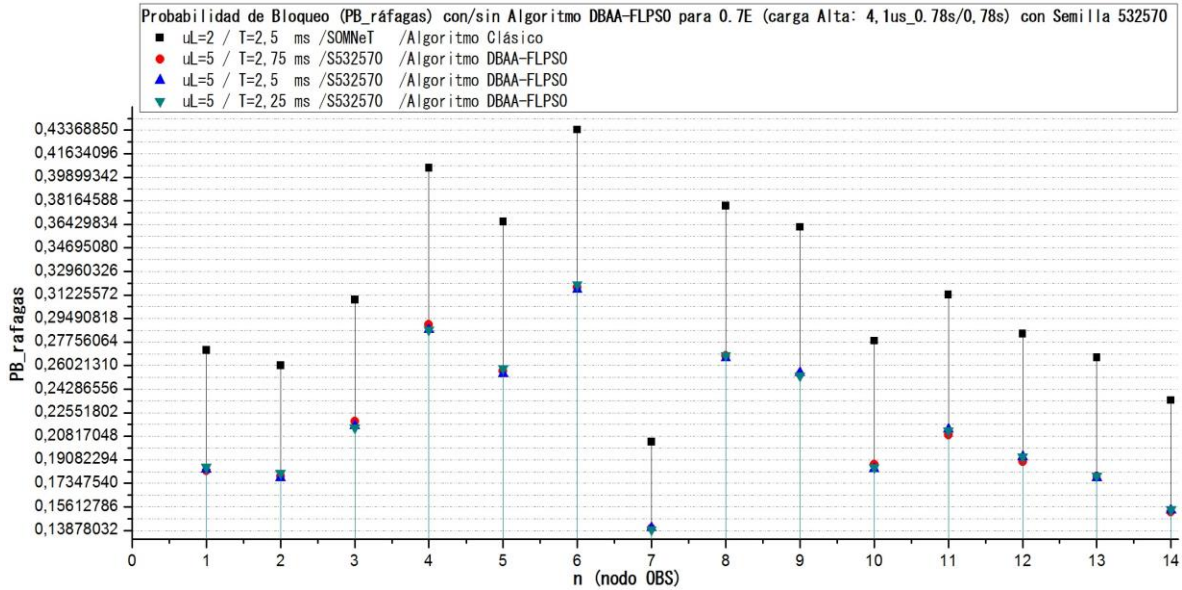


Figura 4.47 PB por nodo para Carga Alta con 0.7E, con $uL=5/T=(2,75,2,25,2,5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2,5$ para el Algoritmo Clásico.

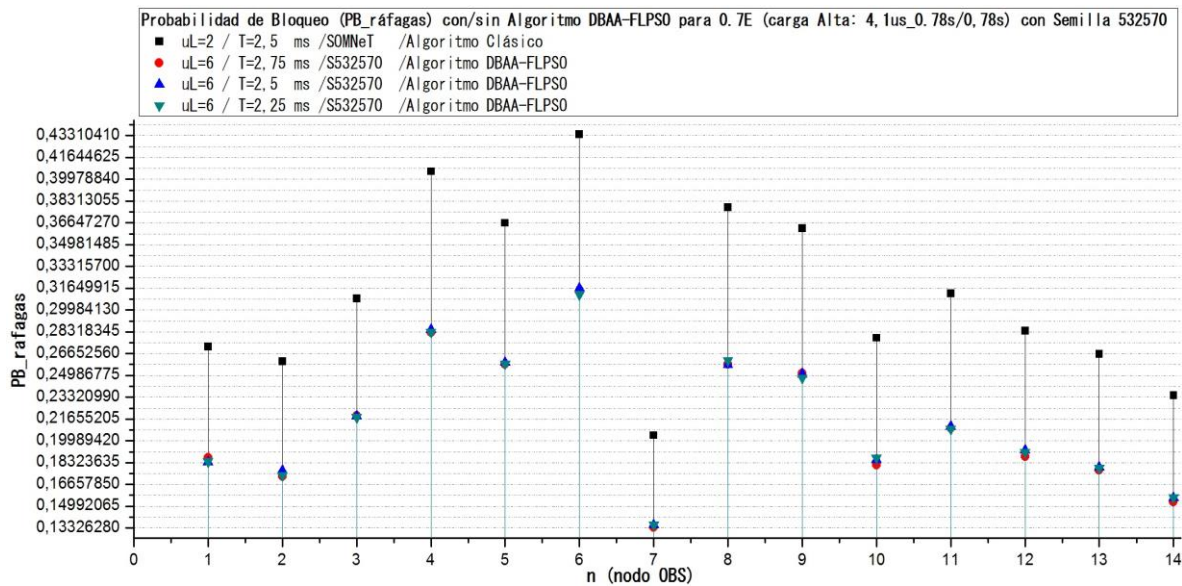


Figura 4.48 PB por nodo para Carga Alta con 0.7E, con $uL=6/T=(2,75,2,25,2,5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2,5$ para el Algoritmo Clásico.

Las figuras 4.46, 4.47 y 4.48, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Alta con 0.7E (4.1µs, 0.78s/0.78s), Semilla 532570 y la combinación de los valores establecidos para uL y T.

Analizando las figuras 4.46, 4.47 y 4.48, en contraste a las figuras para la carga Muy Baja y Carga Baja, con 0.1E, 0.2E, 0.3E y 0.4E respectivamente, para todas las combinaciones uL-T la PB_ráfagas siempre será menor (mejor) al implementar



el algoritmo DMBAA-FLPSO que con el algoritmo de ensamble Clásico. Además, se observa que al aumentar el valor de uL mejora la $PB_{ráfagas}$ en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.49 y la 4.50, muestran el nodo con mayor $PB_{ráfagas}$ (nodo 6) y el nodo con menor $PB_{ráfagas}$ (nodo 7), respectivamente, presentando a mejor escala la $PB_{ráfagas}$ para cada combinación uL y T .

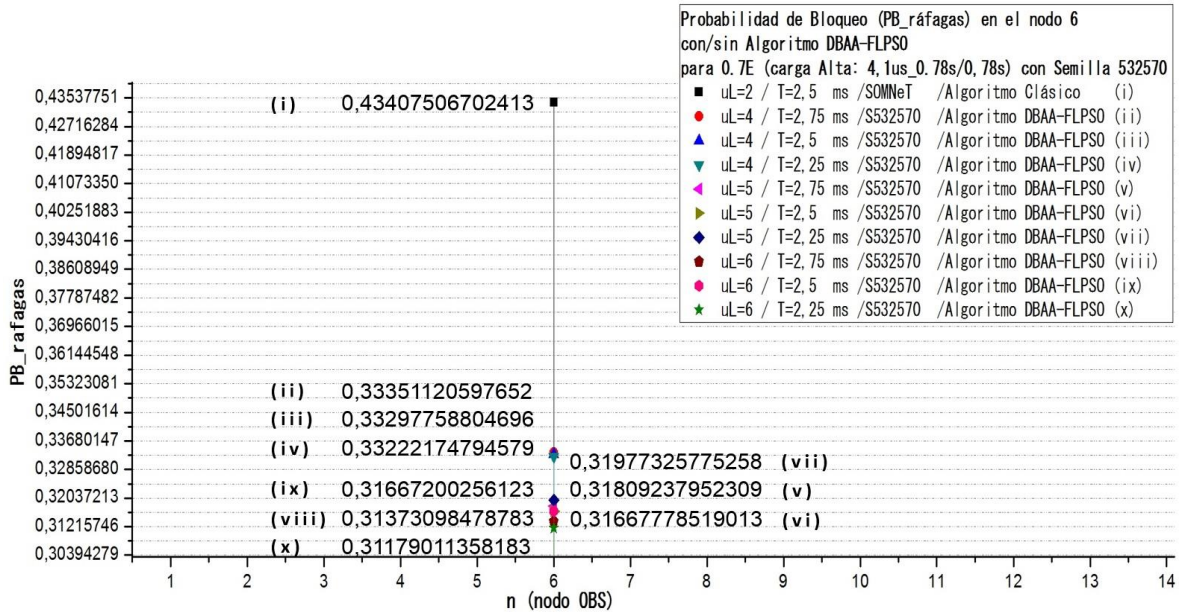


Figura 4. 49 PB nodo 6 para Carga Alta con 0.7E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

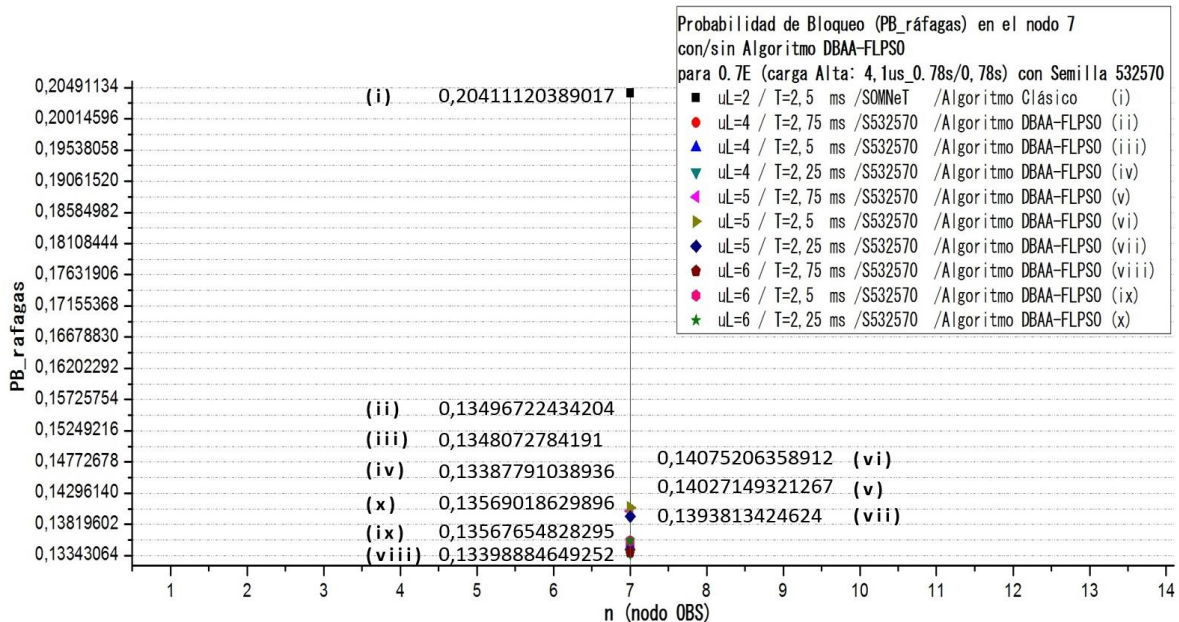


Figura 4. 50 PB nodo 7 para Carga Alta con 0.7E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

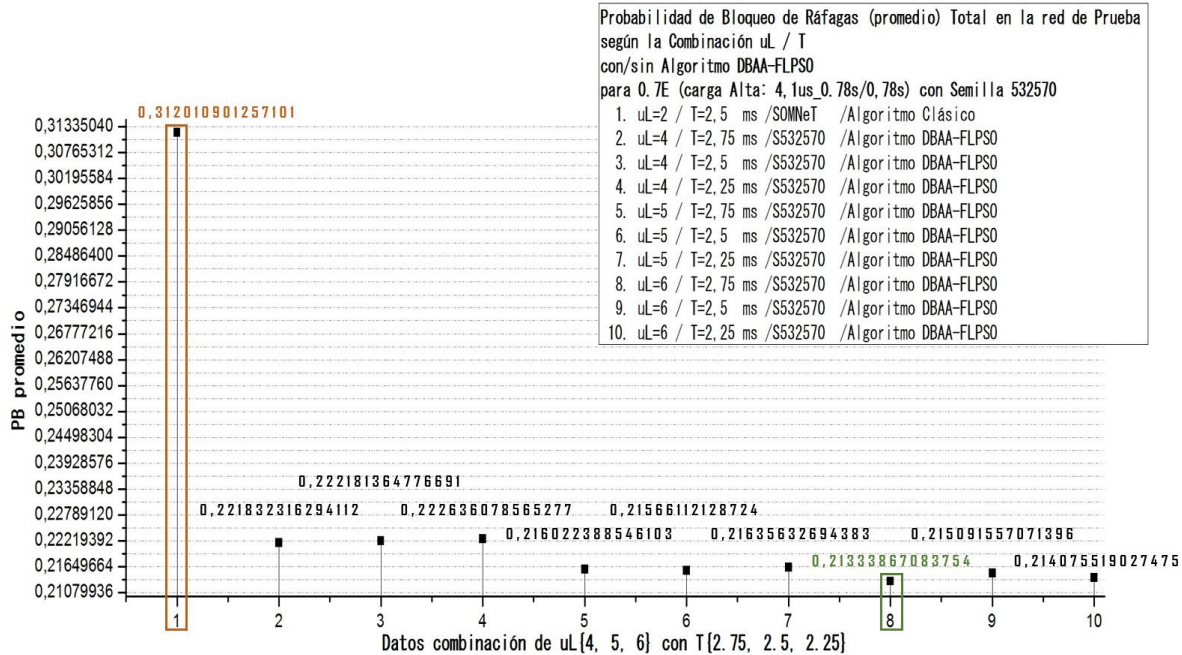


Figura 4. 51 PB promedio por red para Carga Alta con 0.7E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Alta con 0.7E, ver figura 4.51, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.21, la **PB_ráfagas= 0,312010901257101** con **uL=2 y T=2.5ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.51) y la **PB_ráfagas= 0,21333867083754** con **uL=6 y T=2.75ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.51).

Tabla 4. 21 PB_ráfagas para Carga Baja 0.7E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Alta (4.1µs, 0.78s/0.78s)	0.7E	PB_ráfagas= 0,312010901257101 con uL=2 y T=2.5ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,21333867083754 con uL=6 y T=2.75ms. Algoritmo DMBAA-FLPSO



8) Carga Alta para 0.8E (4.0μs, 0.78s/0.78s) con Semilla 150:

Las figuras 4.52, 4.53 y 4.54, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Alta con 0.8E (4.0μs, 0.78s/0.78s), Semilla 150 y la combinación de los valores establecidos para uL y T.

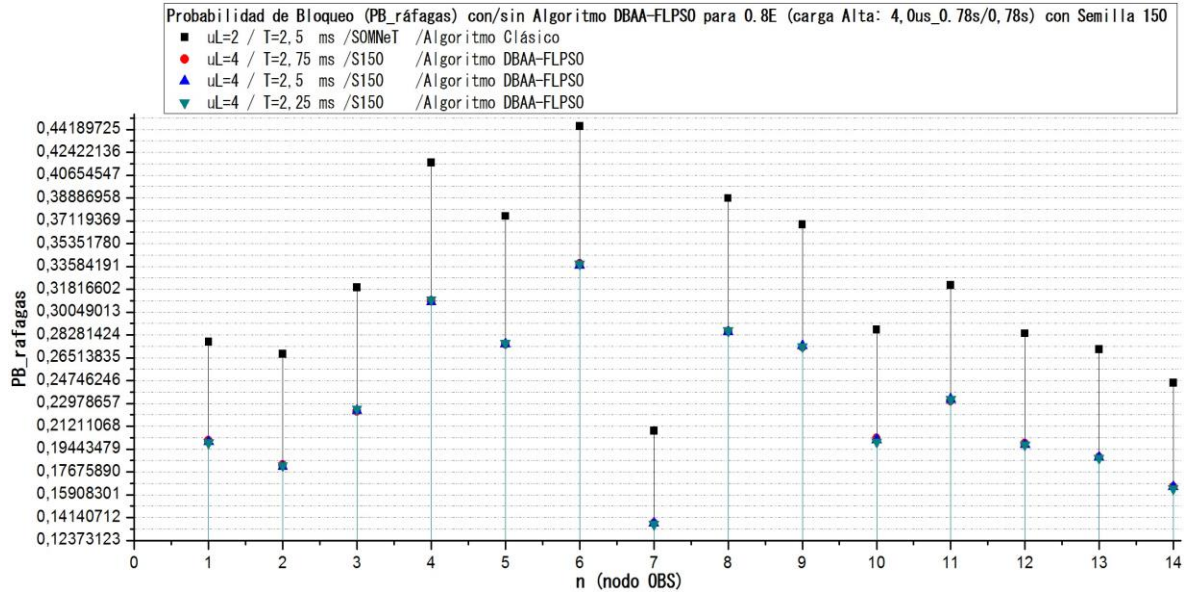


Figura 4. 52 PB por nodo para Carga Alta con 0.8E, con uL=4/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

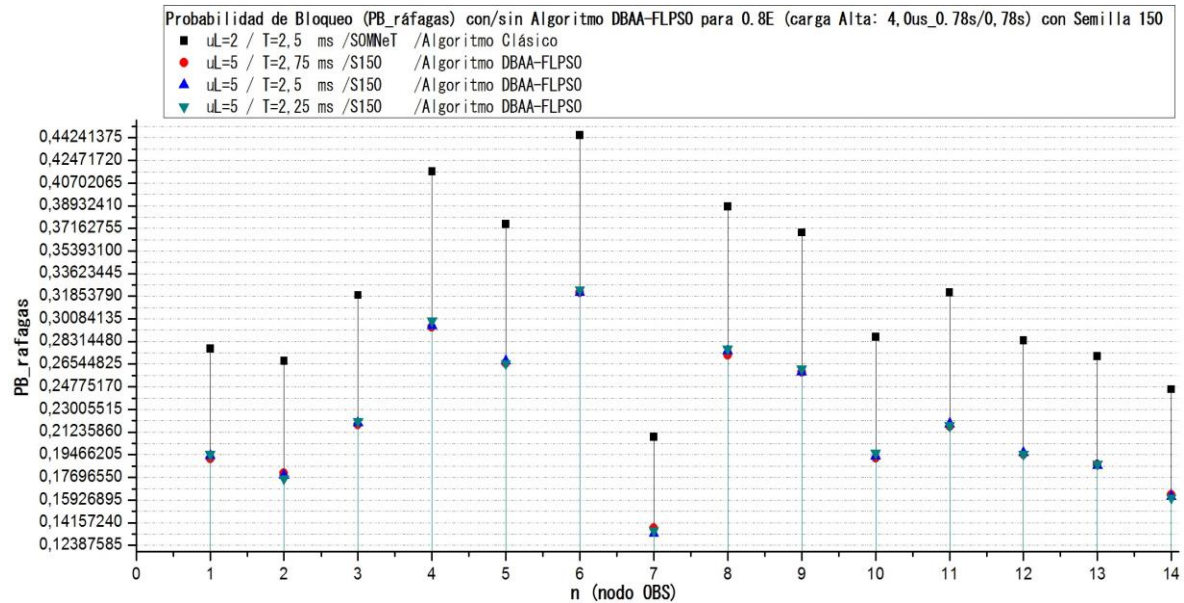


Figura 4. 53 PB por nodo para Carga Alta con 0.8E, con uL=5/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

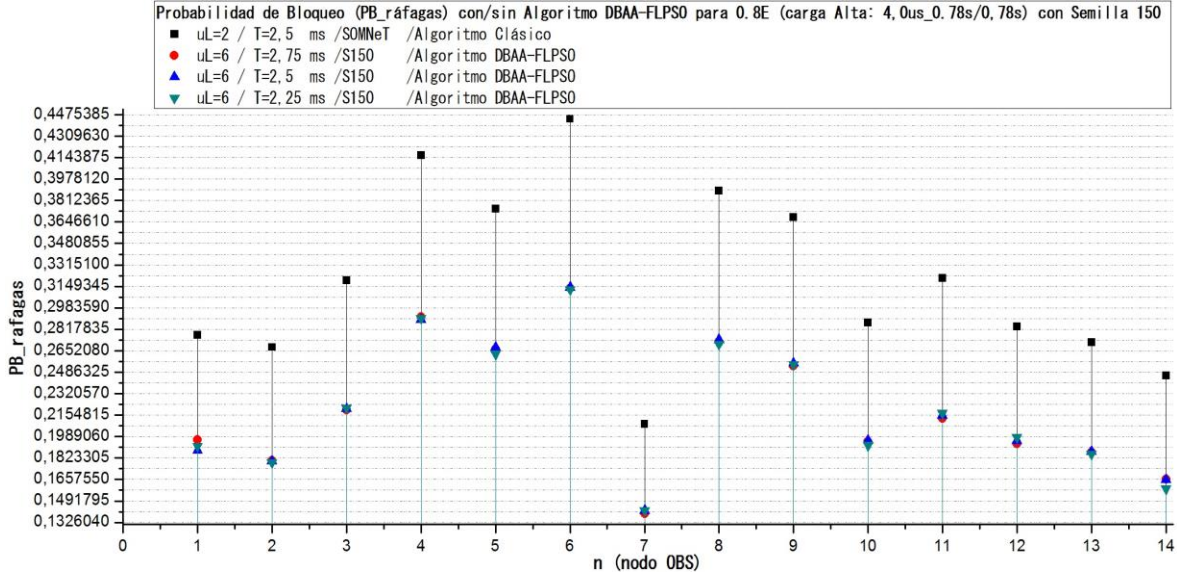


Figura 4. 54 PB por nodo para Carga Alta con 0.8E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.

Analizando las figuras 4.52, 4.53 y 4.54, en contraste a las figuras para la carga Muy Baja y Carga Baja, con 0.1E, 0.2E, 0.3E y 0.4E respectivamente, para todas las combinaciones $uL-T$ la $PB_{ráfagas}$ siempre será menor (mejor) al implementar el algoritmo DBAA-FLPSO que con el algoritmo de ensamble Clásico. Además, se observa que al aumentar el valor de uL mejora la $PB_{ráfagas}$ en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.55 y la 4.56, muestran el nodo con mayor $PB_{ráfagas}$ (nodo 6) y el nodo con menor $PB_{ráfagas}$ (nodo 7), respectivamente, presentando a mejor escala la $PB_{ráfagas}$ para cada combinación uL y T .

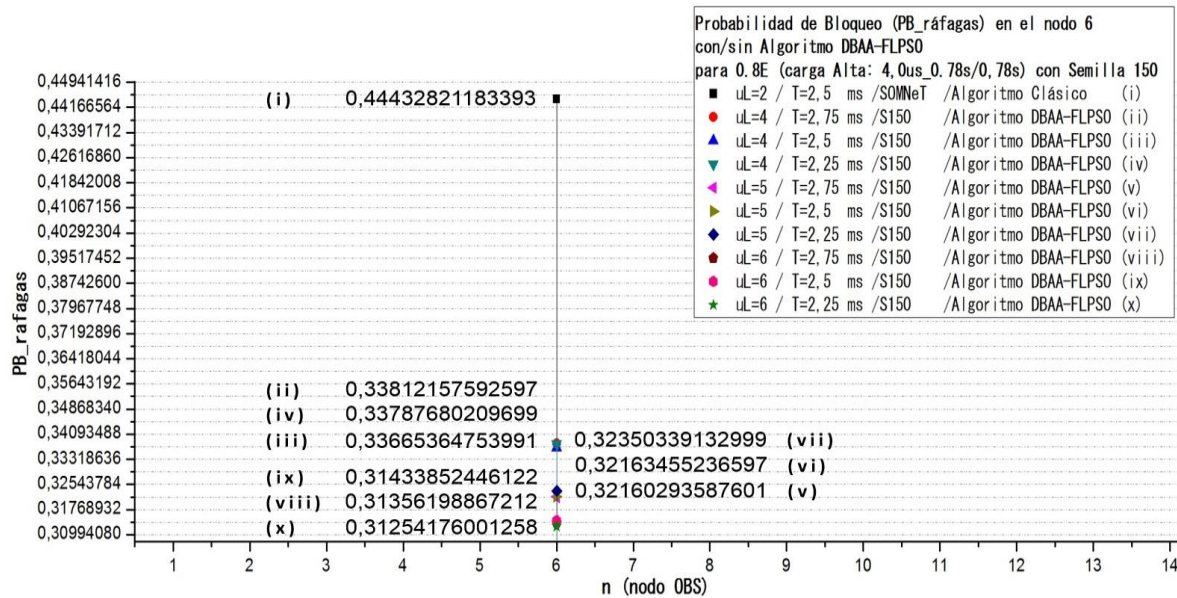


Figura 4. 55 PB nodo 6 para Carga Alta con 0.8E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DBAA-FLPSO y con $uL=2/T=2.5$ para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

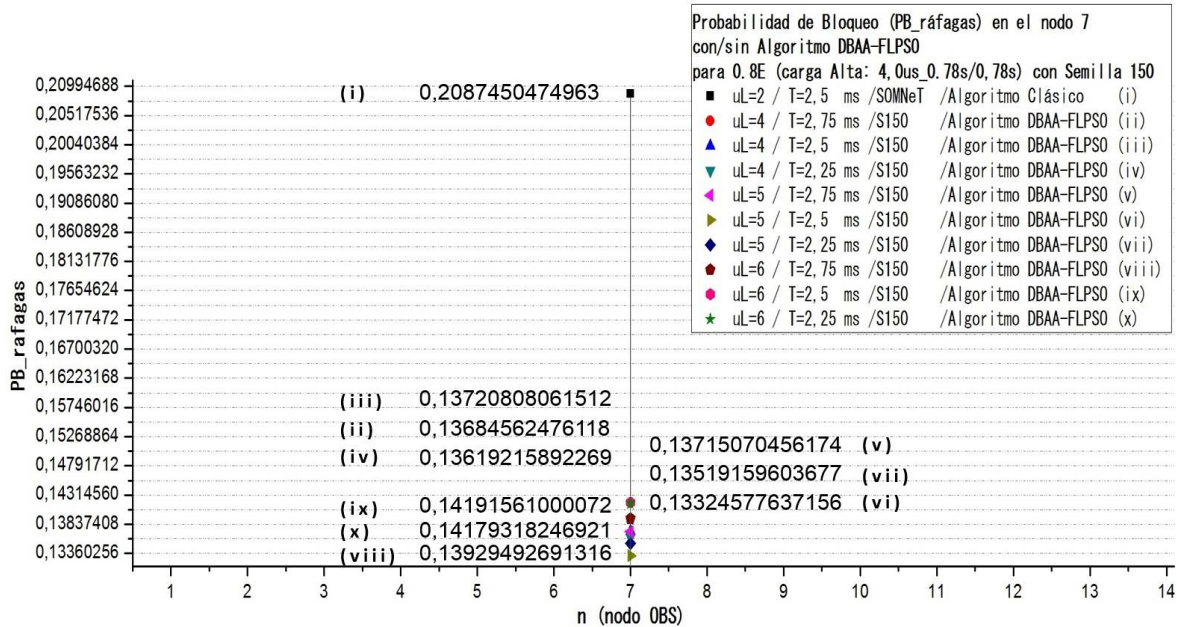


Figura 4. 56 PB nodo 7 para Carga Alta con 0.8E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

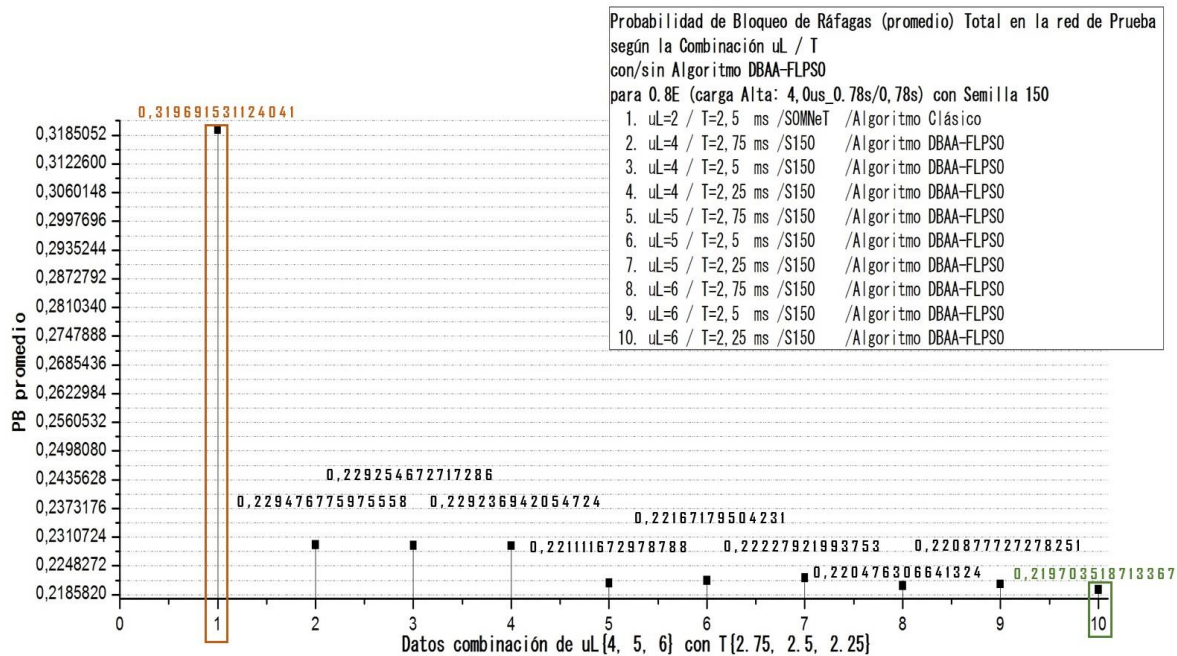


Figura 4. 57 PB promedio por red para Carga Alta con 0.8E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBA-FLPSO y con uL=2/T=2.5 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Alta con 0.8E, ver figura 4.57, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBA-FLPSO. Donde, al implementar el algoritmo DMBA-FLPSO, la PSO realiza una comparación de los



9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.22, la **PB_ráfagas= 0,319691531124041** con **uL=2** y **T=2.5ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.57) y la **PB_ráfagas= 0,219703518713367** con **uL=6** y **T=2.25ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.57).

Tabla 4. 22 PB_ráfagas para Carga Baja 0.8E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Alta (4.0μs, 0.78s/0.78s)	0.8E	PB_ráfagas= 0,319691531124041 con uL=2 y T=2.5ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,219703518713367 con uL=6 y T=2.25ms. Algoritmo DMBAA-FLPSO

4.6 5 Pruebas de Simulación para Carga Muy Alta, sin/con el Algoritmo DMBAA-FLPSO, con Semilla 70 /Semilla 20 y para uL=(4, 5, 6) y T=(2.75, 2.5, 2.25). Determinando PB_ráfagas y e2e-D_ráfagas

9) Carga Alta para 0.9E (3.8μs, 0.78s/1.2s) con Semilla 70:

Las figuras 4.58, 4.59 y 4.60, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Muy Alta con 0.9E (3.8μs, 0.78s/1.2s), Semilla 70 y la combinación de los valores establecidos para uL y T.

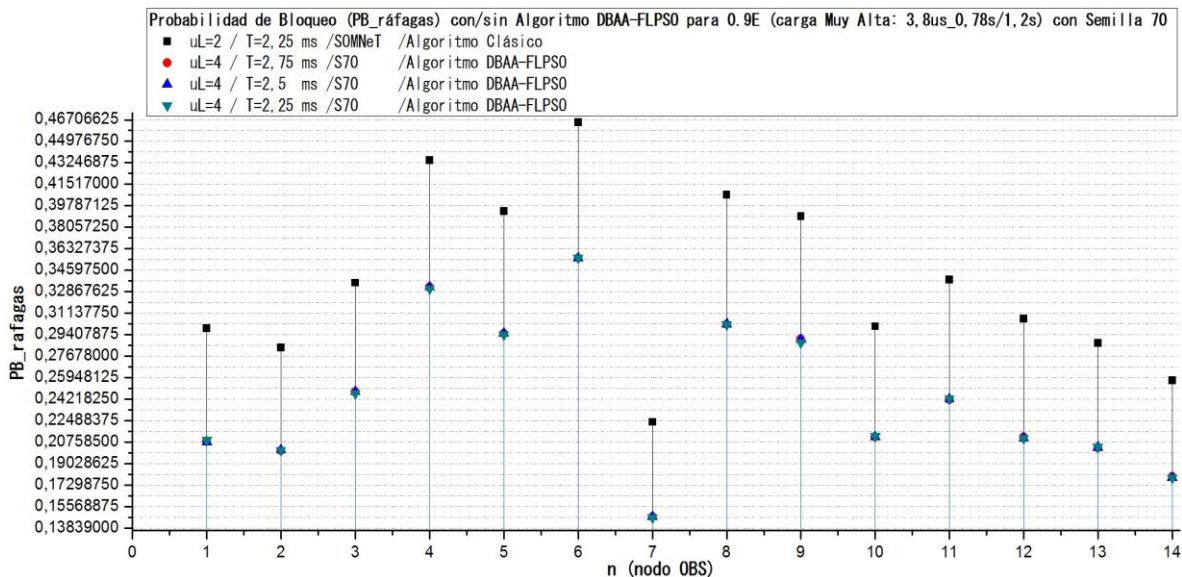


Figura 4. 58 PB por nodo para Carga Muy Alta con 0.9E, con uL=4/T=(2.75,2.25,2.5) para el Algoritmo DMBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

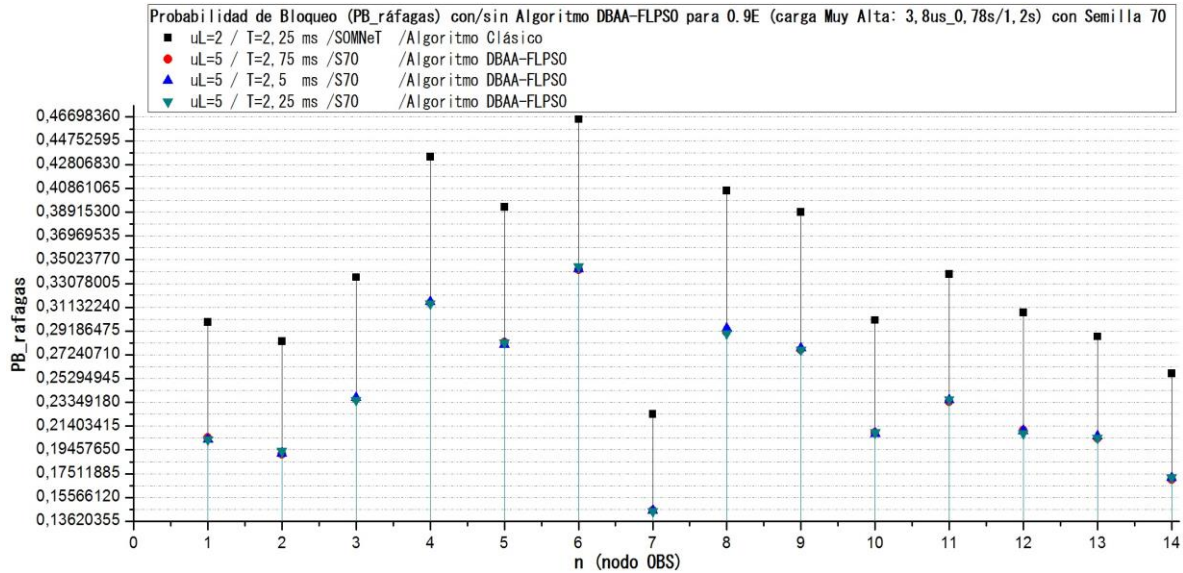


Figura 4. 59 PB por nodo para Carga Muy Alta con 0.9E, con $uL=5/T=(2.75,2.25,2.5)$ para el Algoritmo DMBA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.

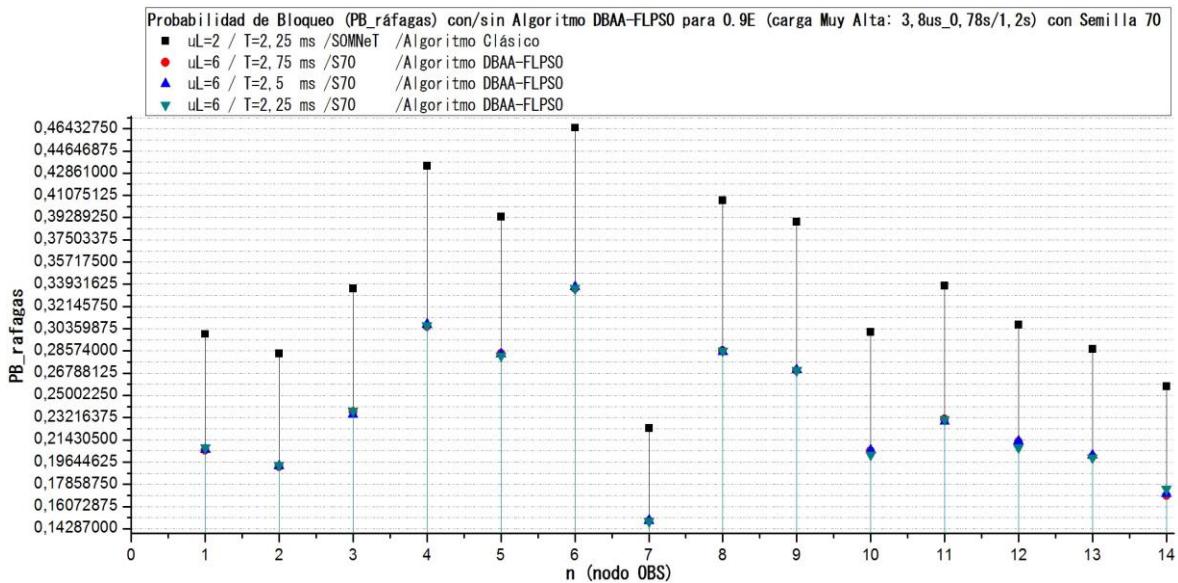


Figura 4. 60 PB por nodo para Carga Muy Alta con 0.9E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DMBA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.

Analizando las figuras 4.58, 4.59 y 4.60, en contraste a las figuras para la carga Muy Baja y Carga Baja, con 0.1E, 0.2E, 0.3E y 0.4E respectivamente, para todas las combinaciones $uL-T$ la $PB_{ráfagas}$ siempre será menor (mejor) al implementar el algoritmo DMBA-FLPSO que con el algoritmo de ensamble Clásico. Además, se observa que al aumentar el valor de uL mejora la $PB_{ráfagas}$ en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.61 y la 4.62, muestran el nodo con mayor $PB_{ráfagas}$ (nodo 6) y el nodo con menor $PB_{ráfagas}$ (nodo 7), respectivamente, presentando a mejor escala la $PB_{ráfagas}$ para cada combinación uL y T .



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

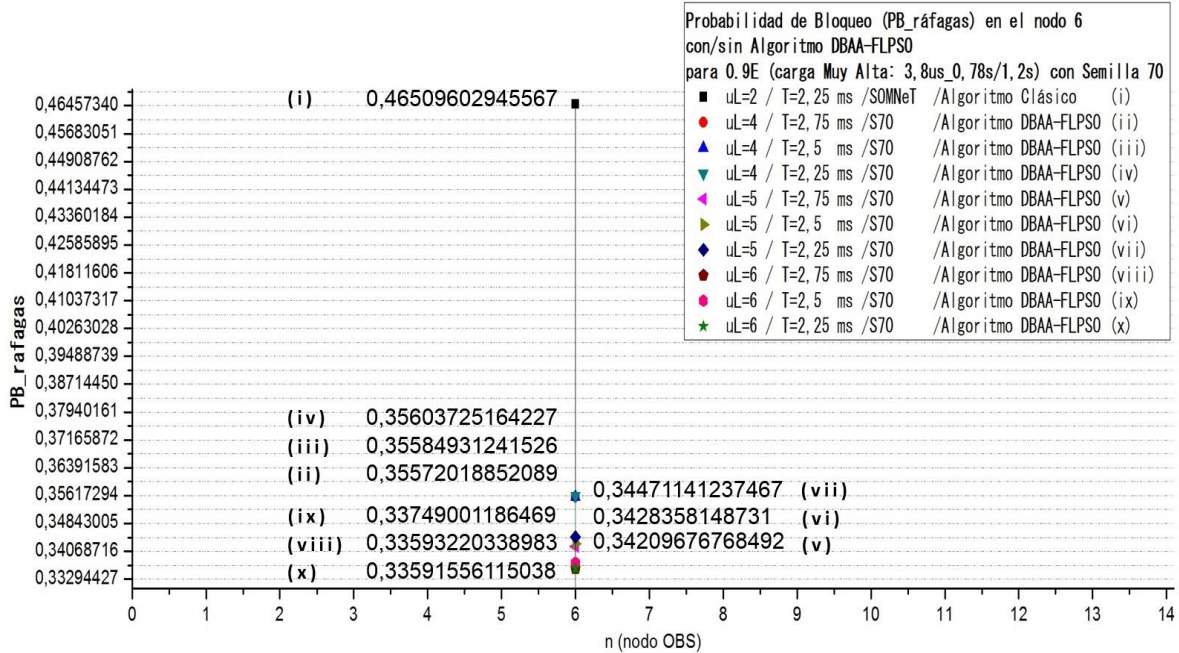


Figura 4. 61 PB nodo 6 para Carga Muy Alta con 0.9E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBAА-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

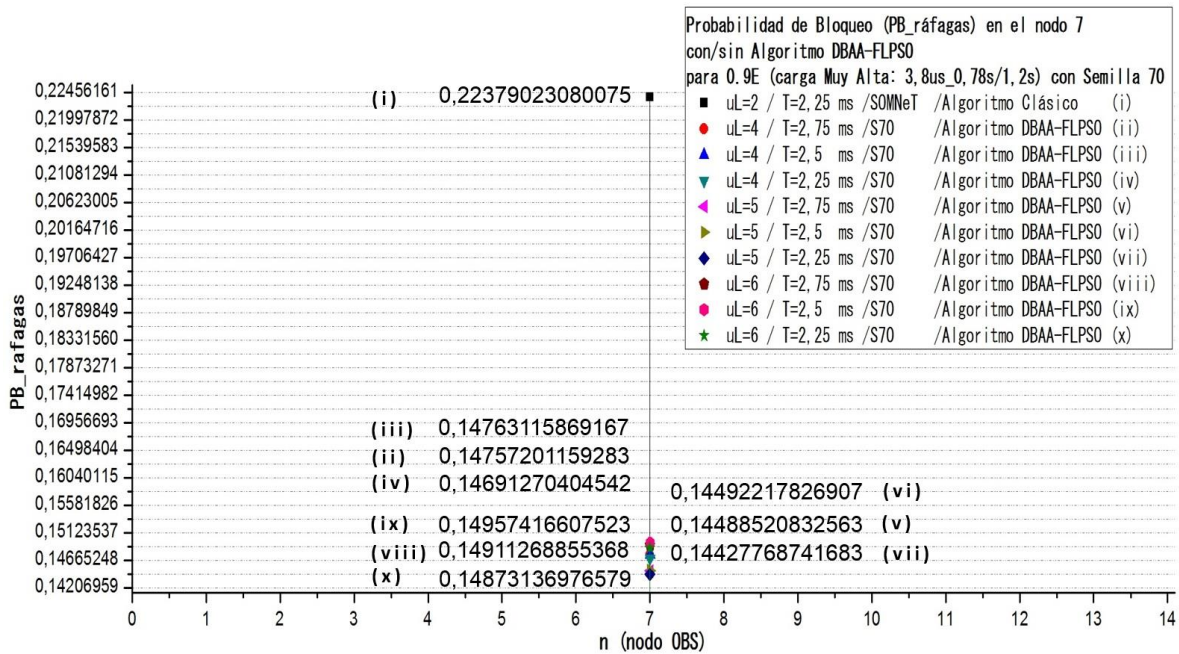


Figura 4. 62 PB nodo 7 para Carga Muy Alta con 0.9E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBAА-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

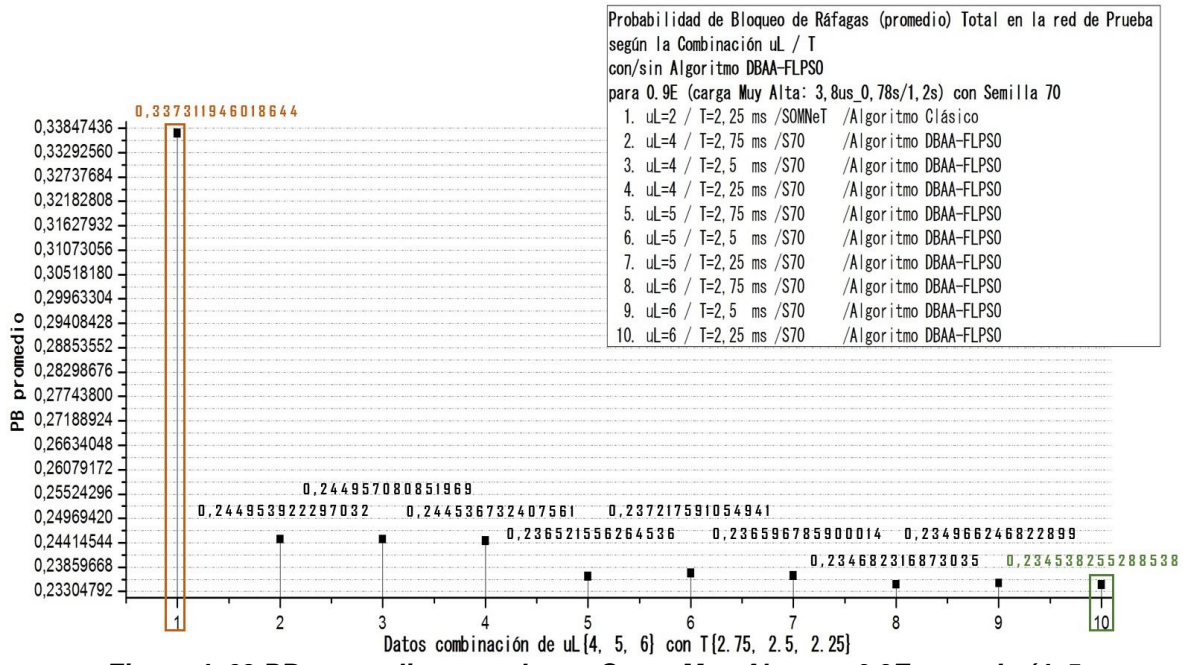


Figura 4. 63 PB promedio por red para Carga Muy Alta con 0.9E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Muy Alta con 0.9E, ver figura 4.63, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBAA-FLPSO. Donde, al implementar el algoritmo DMBAA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.23, la **PB_ráfagas= 0,337311946018644** con **uL=2 y T=2.25ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.63) y la **PB_ráfagas= 0,234538255288538** con **uL=6 y T=2.25ms** al implementar el algoritmo DMBAA-FLPSO (en color verde, ver figura 4.63).

Tabla 4. 23 PB_ráfagas para Carga Baja 0.9E con y sin el algoritmo DMBAA-FLPSO.

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBAA-FLPSO)
Muy Alta (3.8μs, 0.78s/1.2s)	0.9E	PB_ráfagas= 0,337311946018644 con uL=2 y T=2.25ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,234538255288538 con uL=6 y T=2.25ms. Algoritmo DMBAA-FLPSO



10)Carga Alta para 1E (3.7 μ s, 0.78s/1.2s) con Semilla 20:

Las figuras 4.64, 4.65 y 4.66, muestran la probabilidad de bloqueo de ráfagas (PB_ráfagas) por nodo en la red de prueba (OBS-Distribuida) para carga de tráfico Muy Alta con 1E (3.7 μ s, 0.78s/1.2s), Semilla 20 y la combinación de los valores establecidos para uL y T.

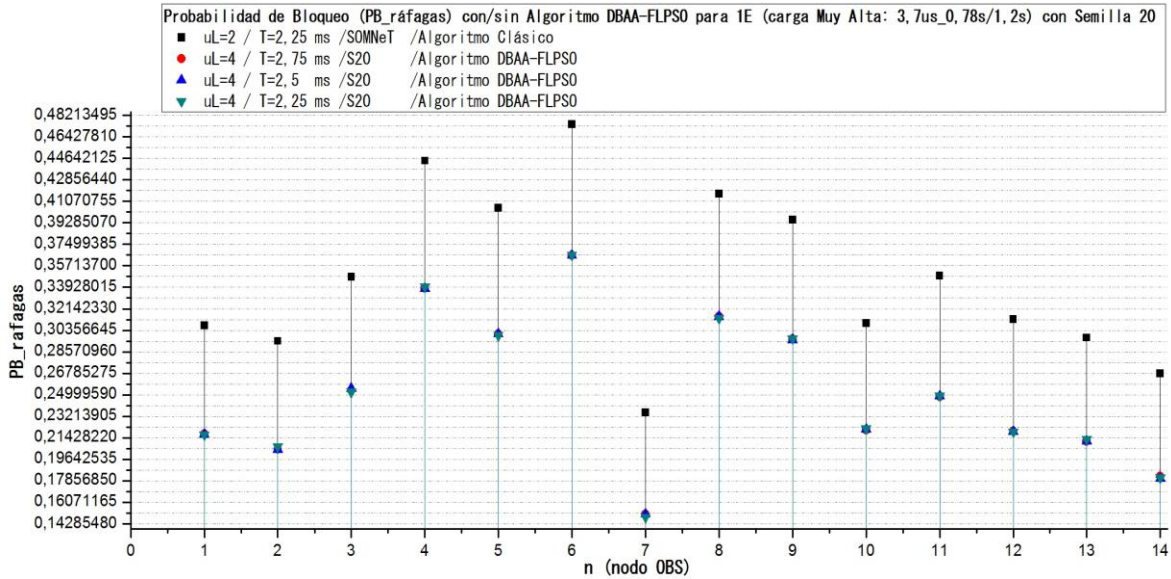


Figura 4. 64 PB por nodo para Carga Muy Alta con 1E, con uL=4/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

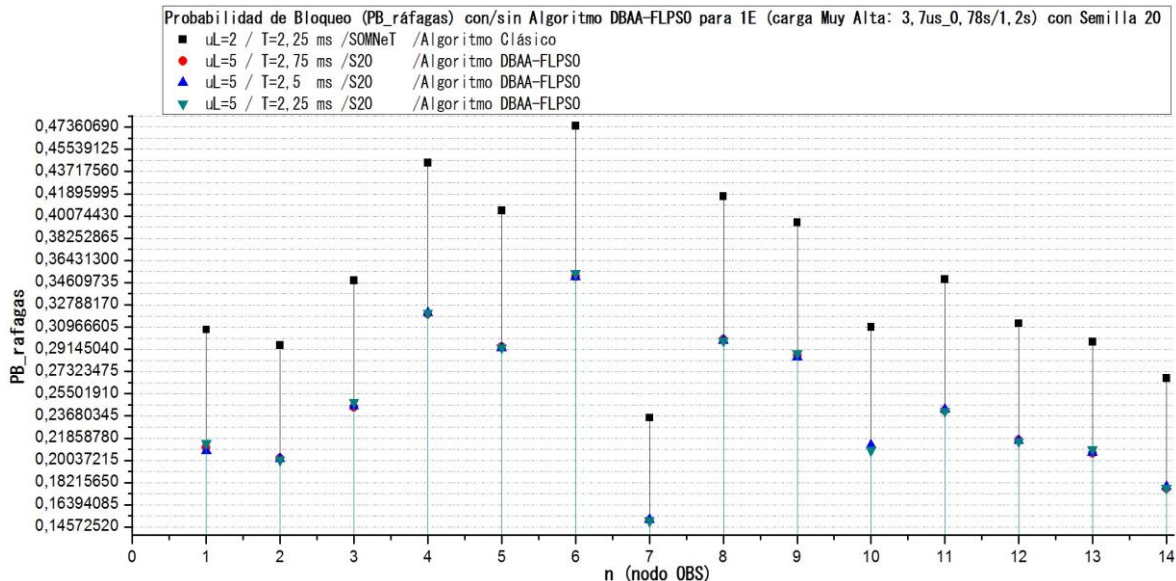


Figura 4. 65 PB por nodo para Carga Muy Alta con 1E, con uL=5/T=(2.75,2.25,2.5) para el Algoritmo DBAA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

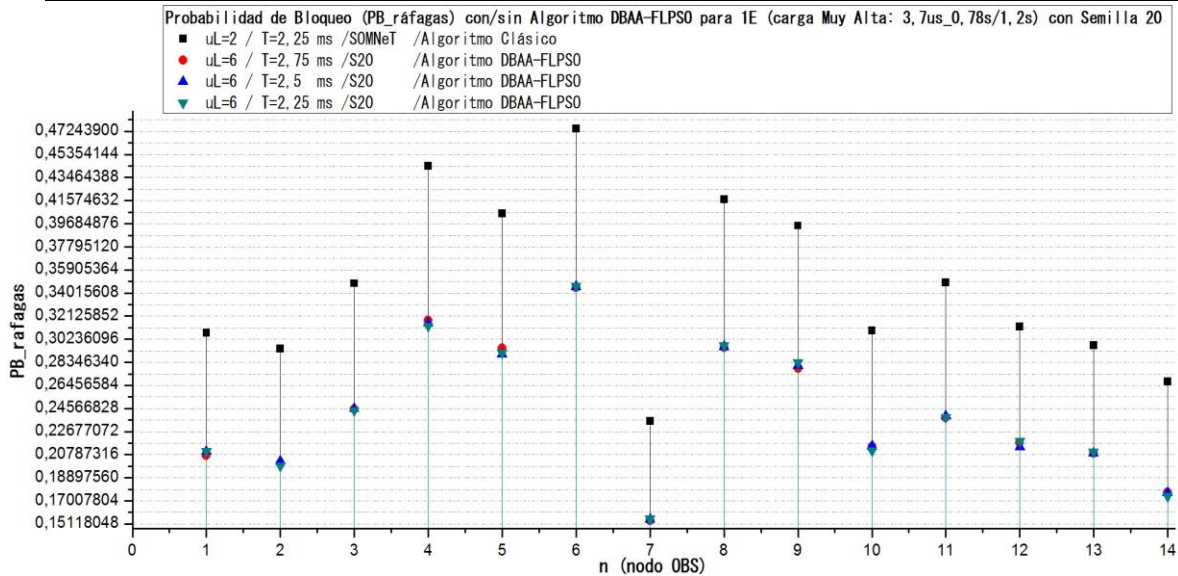


Figura 4. 66 PB por nodo para Carga Muy Alta con 1E, con $uL=6/T=(2.75,2.25,2.5)$ para el Algoritmo DMBA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.

Analizando las figuras 4.64, 4.65 y 4.66, en contraste a las figuras para la carga Muy Baja y Carga Baja, con 0.1E, 0.2E, 0.3E y 0.4E respectivamente, para todas las combinaciones uL-T la PB_ráfagas siempre será menor (mejor) al implementar el algoritmo DMBA-FLPSO que con el algoritmo de ensamble Clásico. Además, se observa que al aumentar el valor de uL mejora la PB_ráfagas en cada uno de los nodos de la red de prueba. Por otro lado, la figura 4.67 y la 4.68, muestran el nodo con mayor PB_ráfagas (nodo 6) y el nodo con menor PB_ráfagas (nodo 7), respectivamente, presentando a mejor escala la PB_ráfagas para cada combinación uL y T.

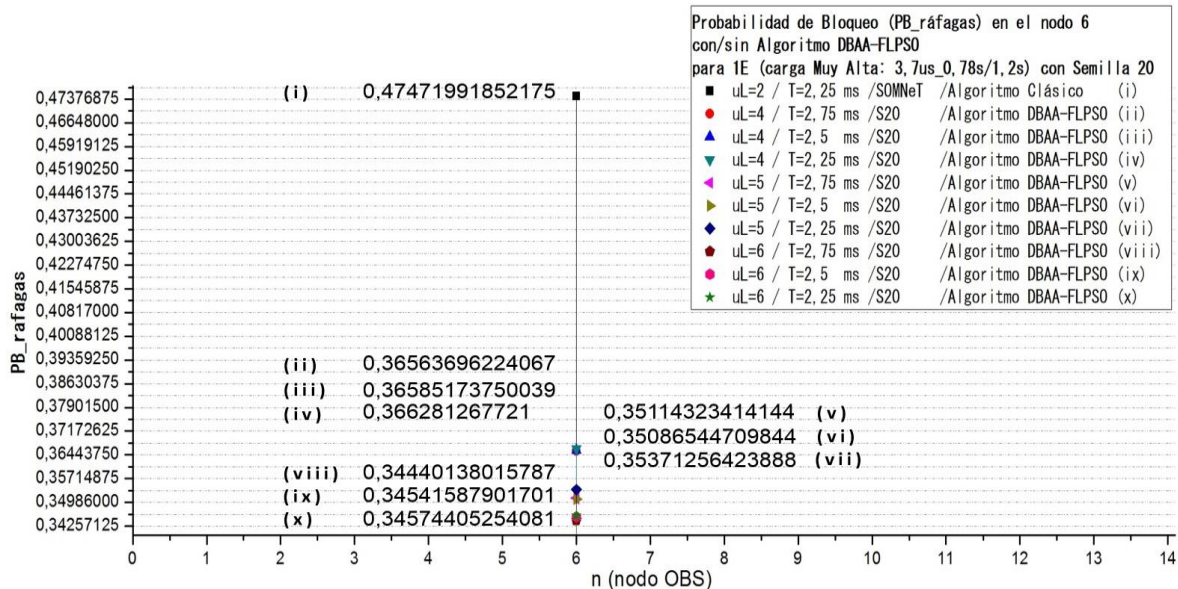


Figura 4. 67 PB nodo 6 para Carga Muy Alta con 1E, con $uL=(4, 5, 6)/T=(2.75,2.25,2.5)$ para el Algoritmo DMBA-FLPSO y con $uL=2/T=2.25$ para el Algoritmo Clásico.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

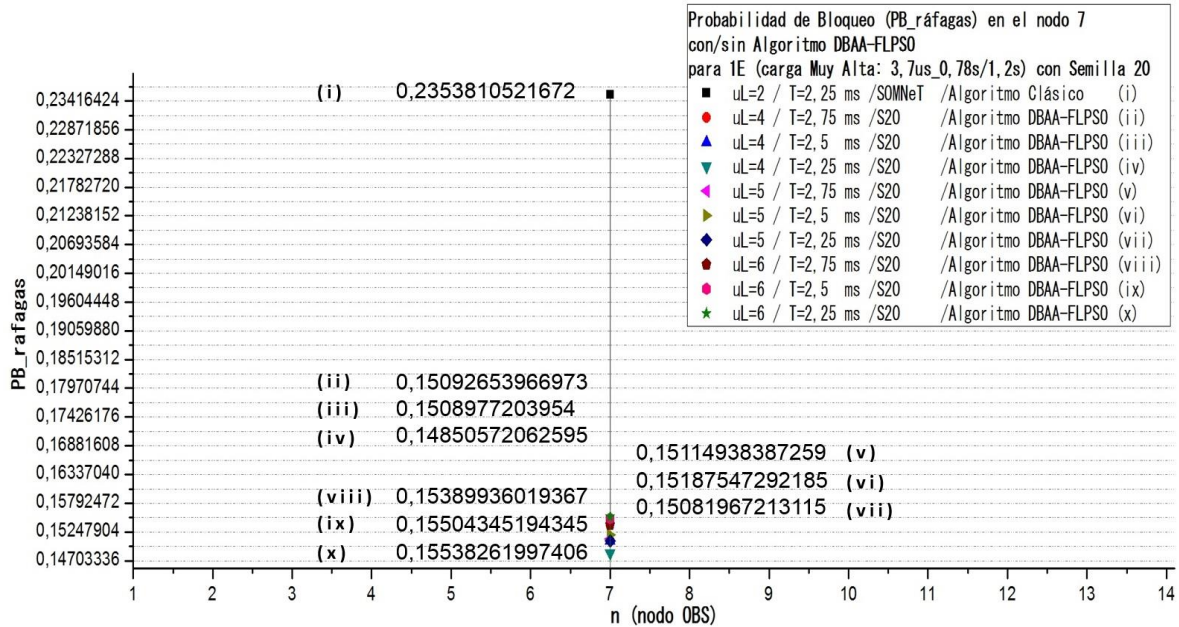


Figura 4. 68 PB nodo 7 para Carga Muy Alta con 1E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

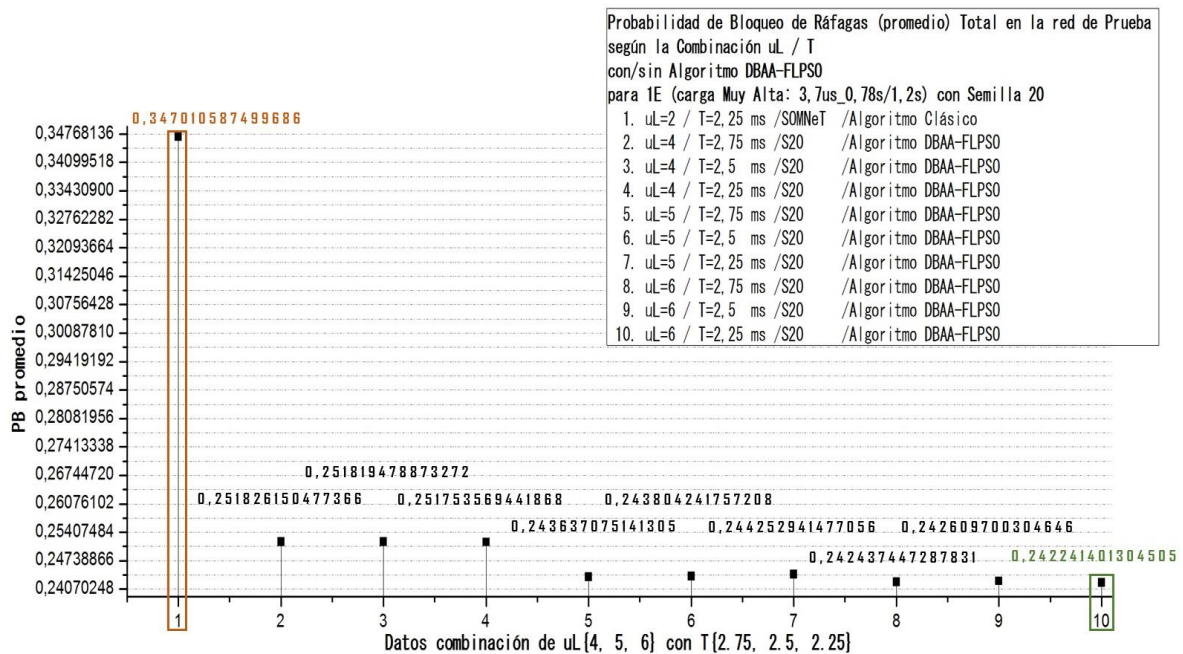


Figura 4. 69 PB promedio por red para Carga Muy Alta con 1E, con uL=(4, 5, 6)/T=(2.75,2.25,2.5) para el Algoritmo DMBA-FLPSO y con uL=2/T=2.25 para el Algoritmo Clásico.

Al finalizar todas las pruebas de simulación para Carga Muy Alta con 1E, ver figura 4.69, se determina la PB_ráfagas total (promedio) al implementar el algoritmo de ensamble Clásico, se escoge la combinación de uL y T con mayor valor (el peor de los casos) y al implementar el algoritmo DMBA-FLPSO. Donde, al



implementar el algoritmo DMBA-FLPSO, la PSO realiza una comparación de los 9 valores dados por la técnica FL determinando la menor PB_ráfagas y su combinación de uL y T. De esta manera, para carga Muy Baja se obtienen dos resultados, ver tabla 4.24, la **PB_ráfagas= 0,347010587499686** con **uL=2 y T=2.25ms** al implementar el algoritmo de ensamble Clásico (en color naranja, ver figura 4.69) y la **PB_ráfagas= 0,242241401304505** con **uL=6 y T=2.25ms** al implementar el algoritmo DMBA-FLPSO (en color verde, ver figura 4.69).

Tabla 4. 24 PB_ráfagas para Carga Baja 1E con y sin el algoritmo DMBA-FLPSO.

Carga de Tráfico		PB_ráfagas con uLy T (Algoritmos de ensamble Clásico) o (Algoritmo DMBA-FLPSO)
Muy Alta (3.7µs, 0.78s/1.2s)	1E	PB_ráfagas= 0,347010587499686 con uL=2 y T=2.25ms. Algoritmo de ensamble Clásico
		PB_ráfagas= 0,242241401304505 con uL=6 y T=2.25ms. Algoritmo DMBA-FLPSO

Con base a las pruebas de simulación realizadas en la red OBS Distribuida (red de prueba), las tablas 4.25, 4.26, 4.27, 4.28 y 4.29 presentan los resultados obtenidos de la probabilidad de bloqueo (PB_ráfagas) y el retardo extremo a extremo (e2e_ráfagas) al implementar el algoritmo de ensamble mixto Clásico y el algoritmo DMBA-FLPSO, variando la carga de tráfico entrante y los valores de los parámetros de ensamble, uL y T.

Tabla 4. 25 PB_ráfagas y e2e-D_ráfagas para Carga Muy Alta.

Carga Volumen del Flujo de Tráfico		uL	T			% PB Algoritmo DBAA-FLPSO	% PB Algoritmo Mixto Clásico			% e2e-D Algoritmo DBAA-FLPSO	% e2e-D Algoritmo Mixto Clásico								
Carga Muy Alta	Dato 10 1E	4	0,00275	25,1826150477	24,2241401305	34,7010587500	0,4054693193	0,4053990581	0,4053990581	0,4053009030									
			0,0025	25,1819478873							25,1753569442	0,4054693350							
			0,00225	25,1753569442							0,4053990581								
		5	0,00275	24,3637075141							24,3637075141	0,4055575049							
			0,0025	24,3804241757								0,4055561643							
			0,00225	24,4252941477								0,4055563000							
		6	0,00275	24,2437447288							24,2241401305	0,4056405289							
			0,0025	24,2609700305								0,4056385429							
			0,00225	24,2241401305								0,4056441892							
		Dato 9 0,9E	4	0,00275							24,4953922297	23,4538255289	33,7311946019	0,4054419239	0,4053980085	0,4053980085	0,4052885444		
				0,0025							24,4957080852							24,4536732408	0,4054417685
				0,00225							24,4536732408							0,4053980085	
	5			0,00275	23,6521556265	23,6521556265	0,4055212910												
				0,0025	23,7217591055		0,4055213163												
				0,00225	23,6596785900		0,4055226135												
	6		0,00275	23,4682316873	23,4538255289	0,4056063372													
			0,0025	23,4966246823		0,4056018957													
			0,00225	23,4538255289		0,4056033426													



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

Tabla 4. 26 PB_ráfagas y e2e-D_ráfagas para Carga Alta.

Carga Volumen del Flujo de Tráfico		uL	T		% PB Algoritmo DBAA-FLPSO	% PB Algoritmo Mixto Clásico		% e2e-D Algoritmo DBAA-FLPSO	% e2e-D Algoritmo Mixto Clásico		
Carga Alta	Dato 8 0,8E	4	0,00275	22,9476775976	22,9236942055	21,9703518713	31,9691531124	0,4053988571	0,4053777571	0,4052636407	
			0.0025	22,9254672717				0,4053992101			
			0,00225	22,9236942055				0,4053998753			
		5	0,00275	22,1111672979				22,1111672979			0,4054696064
			0.0025	22,1671795042							0,4054696885
			0,00225	22,2279219938							0,4054708857
		6	0,00275	22,0476306641				21,9703518713			0,4055375499
			0.0025	22,0877727278							0,4055342149
			0,00225	21,9703518713							0,4055320575
	Dato 7 0,7E		0,00275	22,1832316294	22,1832316294	0,4053230380	0,4053230380				
			0.0025	22,2181364777		0,4053833000					
			0,00225	22,2636078565		0,4053831113					
	5	0,00275	21,6022388546	21,5661121287	0,4054484827	0,4053230380					
		0.0025	21,5661121287		0,4054493502						
		0,00225	21,6356326944		0,4054508603						
		6	0,00275		21,3338670838		21,3338670838	0,4055130554	0,4052537717		
			0.0025		21,5091557071			0,4055162088			
			0,00225		21,4075519027			0,4055130554			

Tabla 4. 27 PB_ráfagas y e2e-D_ráfagas para Carga Media.

Carga Volumen del Flujo de Tráfico		uL	T		% PB Algoritmo DBAA-FLPSO	% PB Algoritmo Mixto Clásico		% e2e-D Algoritmo DBAA-FLPSO	% e2e-D Algoritmo Mixto Clásico		
Carga Media	Dato 6 0,6E	3	0,00275	24,1744632559	24,1463417694	20,4004146088	29,6536835368	0,4052938781	0,4052835477	0,4052371828	
			0.0025	24,1476445671				0,4052937582			
			0,00225	24,1463417694				0,4052835477			
		4	0,00275	20,9672232232				20,9080251420			0,4053487090
			0.0025	20,9080251420							0,4053482056
			0,00225	20,9434656832							0,4053507117
		5	0,00275	20,4004146088				20,4004146088			0,4054091765
			0.0025	20,4533418215							0,4054100963
			0,00225	20,4194362556							0,4054071750
	Dato 5 0,5E		0,00275	23,5374079097	23,5106166957	0,4052846350	0,4052846350				
			0.0025	23,5123338634		0,4052847363					
			0,00225	23,5106166957		0,4052848222					
	4	0,00275	20,5291384613	20,6054925190	0,4053370373						
		0.0025	20,5472607661		0,4053372797						
		0,00225	20,6054925190		0,4053359223						
		5	0,00275		19,6975381256	19,6975381256	0,4052632186	0,4052295608			
			0.0025		19,8098807379		0,4053949037				
			0,00225		19,8294093915		0,4053953242				



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

Tabla 4. 28 PB_ráfagas y e2e-D_ráfagas para Carga Baja.

Carga Volumen del Flujo de Tráfico		uL	T		% PB Algoritmo DBAA-FLPSO	% PB Algoritmo Mixto Clásico		% e2e-D Algoritmo DBAA-FLPSO	% e2e-D Algoritmo Mixto Clásico	
Carga Baja	Dato 4 0,4E	2	0,00275	27,5050297965	27,5050297965	19,3511663859	27,4112463274	0,4052179997	0,4052179996	0,4052173372
			0,0025	27,5050297965				0,4052179996		
		3	0,00225	27,5057079575				0,4052180012		
			0,00275	22,2346762032				0,4052672397		
		0,0025	22,2406183063	0,4052672886						
		0,00225	22,2438331770	0,4052674000						
	Dato 3 0,3E	4	0,00275	19,4071460432	22,2346762032	18,6414195915	26,8225749497	0,4053164061	0,4052131228	0,4052126250
			0,0025	19,4253731173				0,4053156287		
		2	0,00225	19,3511663859				0,4053135916		
			0,00275	26,9014553579				0,4053135916		
		3	0,0025	26,9075174170				0,4052131228		
			0,00225	26,9121942878				0,4052131760		
	Dato 1 0,1E	4	0,00275	21,6212027949	21,5712252997	18,6414195915	24,9330153764	0,4052577282	0,4052772560	0,4051980319
			0,0025	21,6310522062				0,4052577692		
		2	0,00225	21,5712252997				0,4052579537		
			0,00275	18,7873777789				0,4053030275		
		3	0,0025	18,7824966575				0,4053031051		
			0,00225	18,6414195915				0,4053030358		

Tabla 4. 29 PB_ráfagas y e2e-D_ráfagas para Carga Muy Baja.

Carga Volumen del Flujo de Tráfico		uL	T		% PB Algoritmo DBAA-FLPSO	% PB Algoritmo Mixto Clásico		% e2e-D Algoritmo DBAA-FLPSO	% e2e-D Algoritmo Mixto Clásico	
Carga Muy Baja	Dato 2 0,2E	2	0,00275	25,5007407304	25,4871041061	17,6125402701	25,5329195722	0,4052027626	0,4052027306	0,4052015042
			0,0025	25,5007407304				0,4052027626		
		3	0,00225	25,4871041061				0,4052027306		
			0,00275	20,5571767975				0,4052425656		
		0,0025	20,5402155614	0,4052423139						
		0,00225	20,5274413478	0,4052417212						
	Dato 1 0,1E	4	0,00275	17,6125402701	20,5274413478	17,0375620451	24,9330153764	0,4052830647	0,4052828772	0,4051994564
			0,0025	17,7319561242				0,4052828772		
		2	0,00225	17,7651400490				0,4052844573		
			0,00275	24,8419613538				0,4051994564		
		3	0,0025	24,8567448665				0,4051994959		
			0,00225	24,8637489338				0,4051994783		
	Dato 1 0,1E	4	0,00275	20,0366028526	19,9400677534	17,0375620451	24,9330153764	0,4052383542	0,4052772560	0,4051980319
			0,0025	19,9400677534				0,4052385566		
		2	0,00225	19,9910500809				0,4052379888		
			0,00275	17,0769745094				0,4052774756		
		3	0,0025	17,0375620451				0,4052772697		
			0,00225	17,1319564221				0,4052772560		

Por último, con base a los resultados obtenidos en las Pruebas de Simulación, teniendo en cuenta todas combinaciones de los casos y sub casos establecidos, las figuras 4.70, 4.71 y 4.72 presentan los datos-resultados Finales de la Probabilidad de Bloqueo (PB_ráfagas), el Porcentaje de Ráfagas Pérdidas y el



Retardo extremo a extremo (e2e-D_ráfagas), respectivamente, en una red OBS Distribuida específica y mostrando el desempeño del algoritmo DMBAA-FLPSO (propuesto) en la red de prueba en contraste al algoritmo de ensamble Mixto Clásico. Esta actividad es el Objetivo Especifico tres en este trabajo de grado de Maestría.

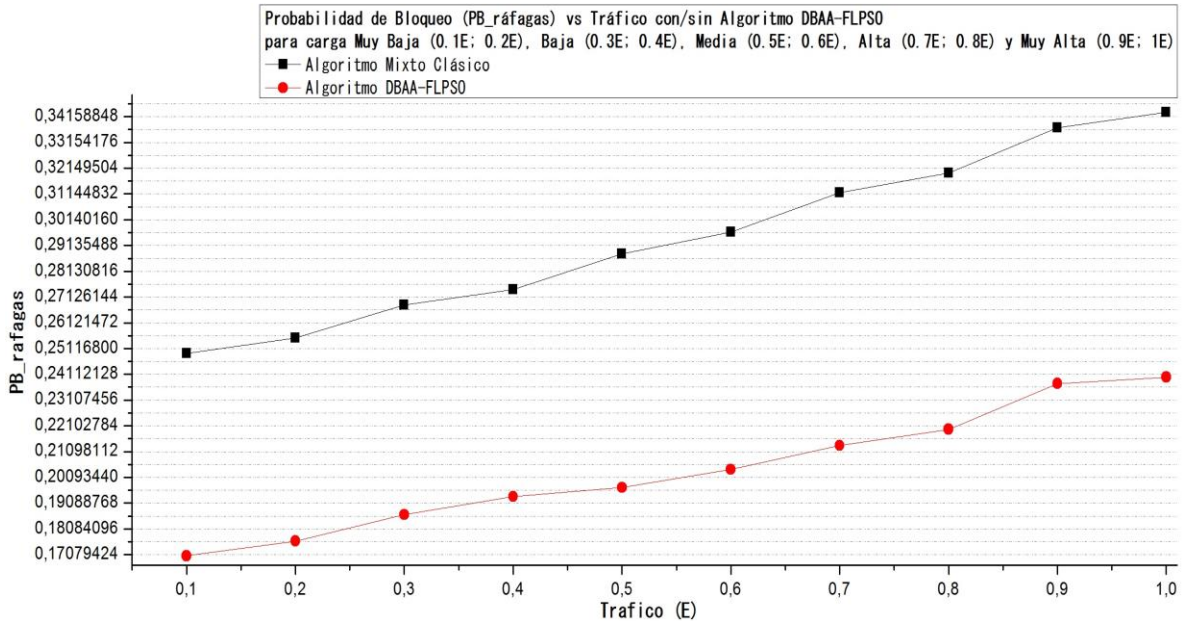


Figura 4. 70 Probabilidad de Bloqueo vs Tráfico para 2.5Gbps Con/Sin Algoritmo DMBAA-FLPSO.

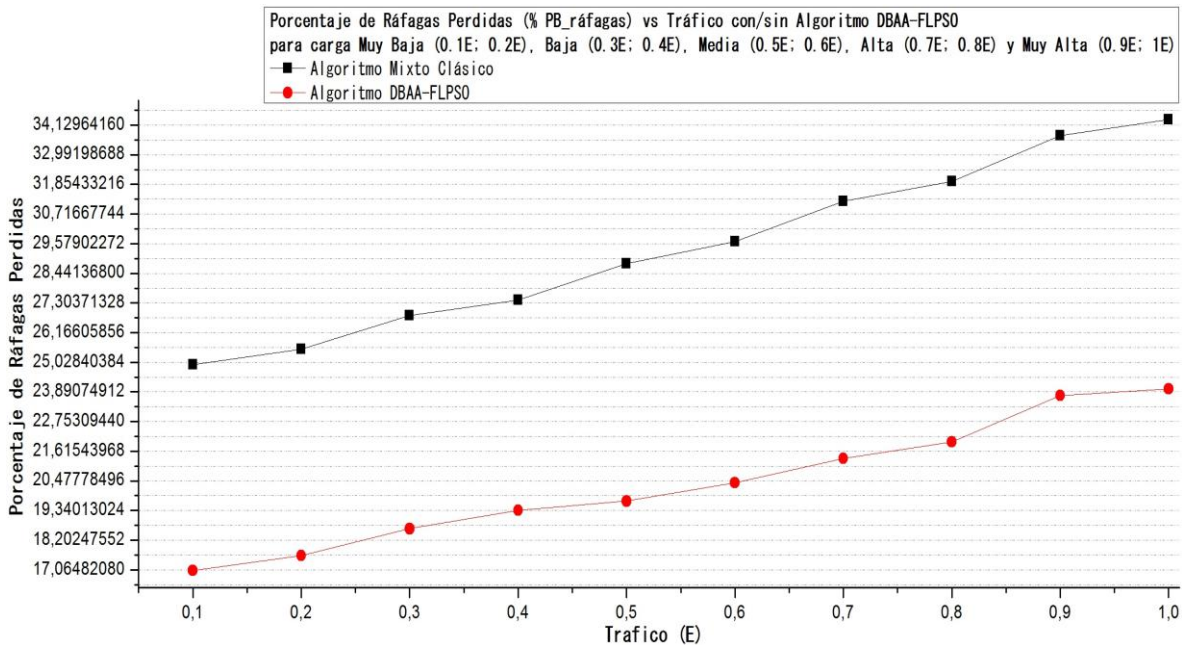


Figura 4. 71 Porcentaje de Ráfagas Perdidas vs Tráfico para 2.5Gbp Con/Sin Algoritmo DMBAA-FLPSO.

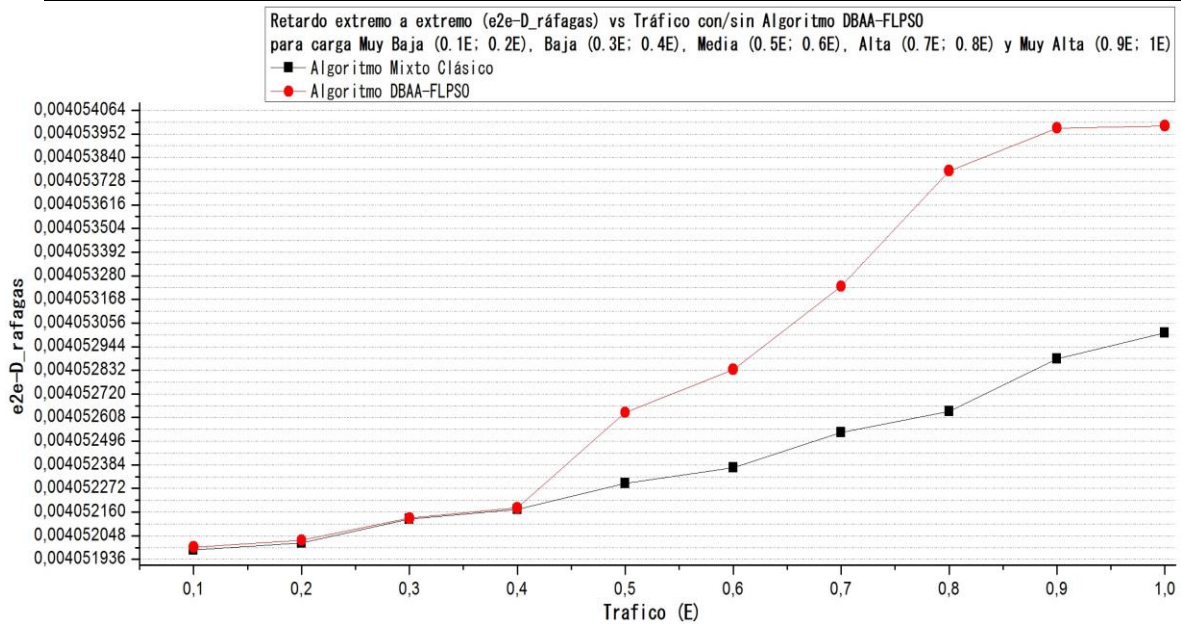


Figura 4. 72 Retardo extremo a extremo (e2e-D_ráfagas) vs Tráfico para 2.5Gbps Con/Sin Algoritmo DMBAA-FLPSO.

Las figuras 4.70 y 4.71 muestran los resultados obtenidos al simular las red OBS Distribuida (red de prueba) al implementar el algoritmo de ensamble Mixto Clásico y el algoritmo DMBAA-FLPSO con 2 canales (landas) por fibra (enlace) y a una velocidad del enlace de 2.5Gbps, como se puede observar, a medida que aumenta el tráfico en las redes, la probabilidad de bloqueo y el porcentaje de ráfagas perdidas se incrementa, esto se debe a que al aumentar el tráfico, la asignación de recursos en la red es cada vez más compleja. La red de prueba con el algoritmo DMBAA-FLPSO presenta un mejor desempeño que la red de prueba con el algoritmo de ensamble Mixto Clásico en términos de probabilidad de bloqueo, dado que el DMBAA-FLPSO determina el valor adecuado de cada parámetro de ensamble, uL y T , y la mejor combinación, para asignar el tamaño óptimo a la ráfaga a transmitir. Además, la red de prueba con el algoritmo DMBAA-FLPSO, tiende a estabilizar la tasa de pérdida de ráfagas al aumentar el tráfico (comportamiento convergente).

Otro de los parámetros de desempeño medidos en este trabajo de grado de Maestría es el retardo extremo a extremo de ráfagas ($e2e_ráfagas$), el cual se define como el tiempo que tarda la ráfaga desde que es ensamblada en el nodo origen hasta el momento en que llega al desensamblador del nodo destino. Como se puede observar en la figura 4.72, el retardo extremo a extremo de ráfagas también se incrementa a medida que el tráfico en la red aumenta, al igual que la probabilidad de bloqueo, esto se da porque al aumentar el número de ráfagas en las colas de los planificadores de envío, los recursos en la red se van agotando de igual manera y las ráfagas tendrán que esperar más tiempo para ser transmitidas, incrementado el retardo. Al analizar el retardo extremo a extremo ($e2e_D_ráfagas$) con el algoritmo DMBAA-FLPSO, en contraste con el algoritmo de ensamble Mixto



Clásico (algoritmo Clásico), no presenta un buen desempeño en la red de prueba, dado que el retardo empeora a medida que aumenta el tráfico. Donde, para carga Muy Baja y Baja (de 0.1E a 0.4E) el e2e-D_ráfagas se mantiene o tiende a ser el mismo valor que con el algoritmo Clásico, por el contrario, para carga media, Alta y Muy Alta (de 0.5E a 1E) la diferencia aumenta. Sin embargo, el algoritmo DMBA-FLPSO para carga Alta y Muy Alta (al aumentar el tráfico) tiende a estabilizar el retardo, presentando un comportamiento convergente, por lo tanto, se obtiene un mejor desempeño en la red al aumentar el tráfico y con el pasar del tiempo.

Las tablas 4.30 y 4.31 presentan una síntesis de los datos-resultados Finales, mostrados en las figuras 4.70, 4.71 y 4.72, con el fin de resumir lo analizado anteriormente en cada uno de ellos.

Tabla 4. 30 Síntesis de los datos-resultados Finales de la PB_ráfagas y el e2e-D para 2.5Gbps al implementar el Algoritmo DMBA-FLPSO y el Algoritmo de Ensamble Mixto Clásico

Tráfico		PB		e2e-D	
		Alg.Clásico Semilla OMNeT	Alg.DMBA- FLPSO Semilla Variable	Alg.Clásico Semilla OMNeT	Alg.DMBA-FLPSO Semilla Variable
Carga Muy Alta	1E 3,7us_0,78s/1,2s	0,3470105874	0,2422414013	0,0040530090	0,0040539905
	0,9E 3,8us_0,78s/1,2s	0,3373119460	0,2345382552	0,0040528854	0,0040539800
Carga Alta	0,8E 4,0us_0,78s/0,78s	0,3196915311	0,2197035187	0,0040526364	0,0040537775
	0,7E 4,1us_0,78s/0,78s	0,3120109012	0,2133386708	0,0040525377	0,0040532303
Carga Media	0,6E 4,3us_0,6s/0,6s	0,296536835	0,2040041460	0,0040523718	0,0040528354
	0,5E 4,4us_0,6s/0,6s	0,2881213134	0,1969753812	0,0040522956	0,0040526321
Carga Baja	0,4E 4,6us_0,4s/0,4s	0,2741124632	0,1935116638	0,0040521733	0,0040521799
	0,3E 4,7us_0,4s/0,4s	0,2682257494	0,1864141959	0,0040521262	0,0040521312
Carga Muy Baja	0,2E 4,9us_0,2/0,2s	0,2553291957	0,1761254027	0,0040520150	0,0040520273
	0,1E 5,0us_0,2/0,2s	0,2493301537	0,1703756204	0,0040519803	0,0040519945



Tabla 4. 31 Síntesis de los datos-resultados Finales en porcentaje de la PB_ráfagas y el e2e-D para 2.5Gbps al implementar el Algoritmo DMBAА-FLPSO y el Algoritmo de Ensamble Mixto Clásico

Tráfico		% PB		% e2e-D	
		Algoritmo Clásico Semilla OMNeT	Algoritmo DMBAА-FLPSO Semilla Variable	Algoritmo Clásico Semilla OMNeT	Algoritmo DMBAА-FLPSO Semilla Variable
Carga Muy Alta	1E 3,7us_0,78s/1,2s	34,7010587499	24,2241401304	0,4053009030	0,4053990581
	0,9E 3,8us_0,78s/1,2s	33,7311946018	23,4538255288	0,4052885444	0,40539800851
Carga Alta	0,8E 4,0us_0,78s/0,78s	31,9691531124	21,9703518713	0,4052636407	0,40537775709
	0,7E 4,1us_0,78s/0,78s	31,2010901257	21,3338670837	0,4052537717	0,40532303797
Carga Media	0,6E 4,3us_0,6s/0,6s	29,6536835367	20,4004146088	0,4052371827	0,40528354773
	0,5E 4,4us_0,6s/0,6s	28,8121313406	19,6975381256	0,4052295607	0,40526321864
Carga Baja	0,4E 4,6us_0,4s/0,4s	27,4112463274	19,3511663858	0,4052173371	0,40521799965
	0,3E 4,7us_0,4s/0,4s	26,8225749496	18,6414195915	0,4052126250	0,40521312275
Carga Muy Baja	0,2E 4,9us_0,2/0,2s	25,5329195721	17,6125402701	0,4052015041	0,40520273059
	0,1E 5,0us_0,2/0,2s	24,9330153764	17,0375620450	0,4051980319	0,40519945637

En la tabla 4.30 y 4.31 se observa una comparación del desempeño en términos de la probabilidad de bloqueo (PB_ráfagas) y el retardo extremo a extremo (e2e_ráfagas) al implementar el algoritmo DMBAА-FLPSO y el algoritmo de ensamble Mixto Clásico, al variar los valores de los parámetros de ensamble, el umbral de longitud (uL) y el tiempo de ensamblaje (T), en un rango determinado (conjunto de valores difusos) y definido como óptimo sobre un tráfico variable (carga Muy Baja, carga Baja, carga Media, carga Alta y carga Muy Alta) y normalizado en Erlang (de 0.1 hasta 1). En la tabla 4.31, se puede observar que la red de prueba que implementa el algoritmo DMBAА-FLPSO presenta una probabilidad de bloqueo más baja que la que implementa el algoritmo Clásico en todo sentido. Por ejemplo, al comparar los resultados al implementar el algoritmo DMBAА-FLPSO con el algoritmo de ensamble Mixto Clásico para carga Muy Baja en 0.1E la PB_ráfagas disminuye a un 17.04% (con DMBAА-FLPSO) en contraste a un 24.93% (con Clásico) y en 0.2E la PB_ráfagas disminuye a un 17.61% (con DMBAА-FLPSO) en contraste a un 25.53% (con Clásico). En conclusión, es mejor



el desempeño de la red de prueba al implementar el algoritmo DMBAА-FLPSO dado que disminuye la PB_ráfagas en un 7.89% en 0.1E y en un 7.92% en 0.2E.

Con base a la tabla 4.31, en la tabla 4.32 y en la figura 4.73, se presentan la diferencia en porcentaje de la PB_ráfagas entre el algoritmo DMBAА-FLPSO y el algoritmo de ensamble Mixto Clásico al variar la carga de tráfico de 0.1E a 1E. De manera que, en la figura 4.32, se puede observar el nivel de desempeño del algoritmo DMBAА-FLPSO en cuanto a la PB_ráfagas, teniendo en cuenta, que al aumentar la diferencia en el porcentaje el desempeño es mejor.

Tabla 4. 32 Diferencia en porcentaje de la PB_ráfagas entre el algoritmo DMBAА-FLPSO y el algoritmo de ensamble Clásico

Tráfico		Diferencia del % PB	
		%PB (Clásico) – %PB (DMBAА-FLPSO)	
		Nivel de Desempeño	
Carga Muy Alta	1E 3,7us_0,78s/1,2s	10.48	
	0,9E 3,8us_0,78s/1,2s	10.28	
Carga Alta	0,8E 4,0us_0,78s/0,78s	10	
	0,7E 4,1us_0,78s/0,78s	9.87	
Carga Media	0,6E 4,3us_0,6s/0,6s	9.25	
	0,5E 4,4us_0,6s/0,6s	9.11	
Carga Baja	0,4E 4,6us_0,4s/0,4s	8.06	
	0,3E 4,7us_0,4s/0,4s	8.18	
Carga Muy Baja	0,2E 4,9us_0,2/0,2s	7.92	
	0,1E 5,0us_0,2/0,2s	7.89	

Por otro lado, con base a la tabla 4.31, en la tabla 4.33 y en la figura 4.74, se presenta la diferencia en porcentaje del e2e-D_ráfagas entre el algoritmo DMBAА-FLPSO y el algoritmo de ensamble Mixto Clásico al variar la carga de tráfico de 0.1E a 1E. Donde, la figura 4.32 muestra el nivel de desempeño del algoritmo DMBAА-FLPSO en cuanto al e2e-D_ráfagas, teniendo en cuenta, que al disminuir la diferencia en el porcentaje el desempeño no es el mejor.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA

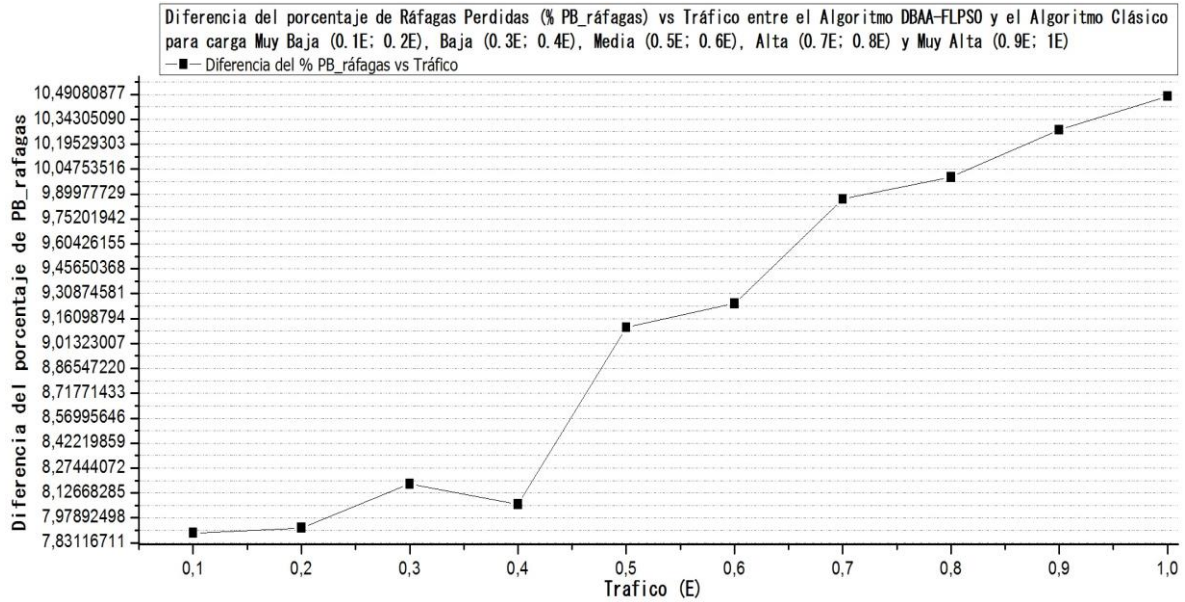


Figura 4. 73 Nivel de Desempeño del Algoritmo DMBAА-FLPSO en cuanto a la PB_ráfagas con un enlace de 2.5Gbps.

Tabla 4. 33 Diferencia en porcentaje del e2e-D_ráfagas entre el algoritmo DMBAА-FLPSO y el algoritmo de ensamble Clásico

Tráfico		Diferencia del % e2e-D_ráfagas
		%PB (Clásico) – %PB (DMBAА-FLPSO)
		Nivel de Desempeño
Carga Muy Alta	1E 3,7us_0,78s/1,2s	-0,0000981550921629
	0,9E 3,8us_0,78s/1,2s	-0,0001094640789980
Carga Alta	0,8E 4,0us_0,78s/0,78s	-0,0001141163644950
	0,7E 4,1us_0,78s/0,78s	-0,0000692662351590
Carga Media	0,6E 4,3us_0,6s/0,6s	-0,0000463649394331
	0,5E 4,4us_0,6s/0,6s	-0,0000336578584160
Carga Baja	0,4E 4,6us_0,4s/0,4s	-0,0000006624959900
	0,3E 4,7us_0,4s/0,4s	-0,0000004977213360
Carga Muy Baja	0,2E 4,9us_0,2/0,2s	-0,0000012264170700
	0,1E 5,0us_0,2/0,2s	-0,0000014244548640



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

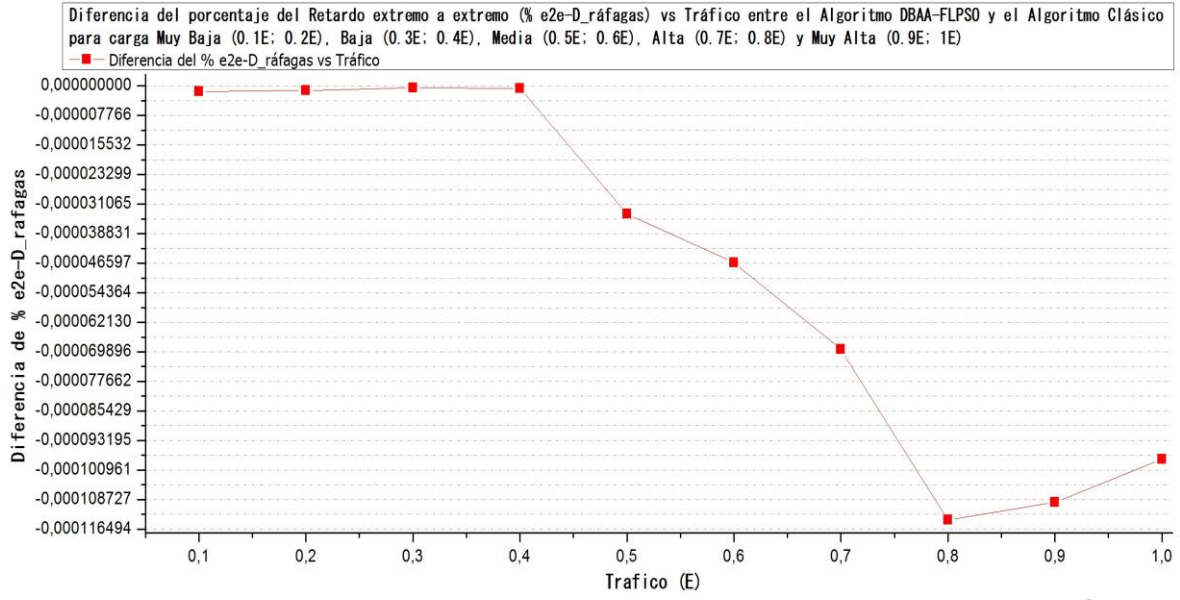


Figura 4.74 Nivel de Desempeño del Algoritmo DMBA-FLPSO en cuanto a la PB_ráfagas con un enlace de 2.5Gbps



CAPÍTULO 5 CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

En este capítulo se presentan las conclusiones, recomendaciones y trabajos futuros que surgen del desarrollo del trabajo de grado. Las conclusiones se presentan respecto a los resultados de simulación, al desarrollo del trabajo de grado y a la herramienta de simulación. Las recomendaciones abarcan aspectos relacionados con la metodología del trabajo y a la herramienta de simulación. Finalmente, los trabajos futuros se presentan respecto a la red OBS Distribuida (red de prueba) y al algoritmo propuesto (DMBAA-FLPSO).

5.1 CONCLUSIONES

5.1.1 Respecto a los resultados de simulación

- Bajo las configuraciones establecidas en la red OBS Distribuida (red de prueba) y de acuerdo a los resultados obtenidos, en las pruebas de simulación y los datos-resultados Finales, el algoritmo DMBAA-FLPSO tiene un mejor desempeño sobre la red de prueba en comparación al algoritmo de ensamble Mixto Clásico, dado que presenta los valores más bajos de probabilidad de bloqueo de ráfagas (PB_rafagas) o tasa de pérdidas de ráfagas, ver figura 4.71. DMBAA-FLPSO, optimiza el desempeño total de la red para cualquier nivel de carga que se le inyecte, donde el nivel de desempeño (valor aproximado) va desde un 8% para carga Baja/Muy Baja, un 9.1% para carga Media hasta un 10.2% para carga Alta/Muy Alta. Además, su comportamiento tiene una tendencia a estabilizarse en un punto (converge), soportando el peor de los escenarios con una carga Muy Alta o máxima, para 1E.
- El algoritmo DMBAA-FLPSO sobre la red de prueba, en cuanto al retardo extremo a extremo por ráfaga (e2e-D), tiene un mejor desempeño cuando se implementa a bajas cantidades de tráfico, para carga Muy Baja (con 0.1E y 0.2E) y para carga Baja (con 0.3E y 0.4E), dado que presenta un comportamiento muy próximo al del algoritmo de ensamble Mixto Clásico. Sin embargo, para carga Media, Alta y Muy Alta (de 0.5E a 1E) el retardo es mayor, pero, su comportamiento tiene una tendencia a estabilizarse en un punto (converge). Este incremento en el retardo, se debe a que el algoritmo propuesto es más robusto por lo tanto es más complejo, dado que es híbrido y adaptativo, donde, requiere más tiempo de procesamiento para determinar un conjunto de valores difusos (óptimo), al utilizar la técnica Lógica Difusa (FL) para cada parámetro de ensamble, uL y T , y otro tiempo para



determinar por comparación el valor adecuado para uL y T , al implementar la PSO. Ver tablas de la 4.25 a la 4.29.

- Bajo las configuraciones establecidas en la red OBS Distribuida (red de prueba) y con base a los resultados obtenidos en las pruebas de simulación, ver figuras 4.10 a la 4.69, el algoritmo DMBAA-FLPSO mejora el desempeño en la red de prueba cuando el valor del parámetro de ensamble uL es el más grande dentro del conjunto determinado, por la técnica FL, para cada combinación de los casos y sub casos (diferentes combinaciones de uL y T). Donde, el valor de uL tiene mayor influencia en el desempeño total de la red solo para carga Media, Alta y Muy Alta. En cambio, para carga Baja y Muy Baja, el valor del parámetro de ensamble T es el que tiene mayor influencia sobre el desempeño total de la red.

5.1. 2 Respecto al desarrollo del Trabajo de Grado de Maestría

- Este trabajo de grado de Maestría se caracteriza por ser un aporte para el grupo de investigación I+D Nuevas Tecnologías en Telecomunicaciones – GNTT de la Universidad del Cauca, ya que puntualmente el tema de investigación, el problema, el ensamble de ráfagas de datos en una red OBS Distribuida, es una extensión de una Tesis de Doctoral de la GNTT, desarrollada por el Ph.D José Giovanni López Perafán.
- Para el desarrollo de este trabajo de grado de Maestría, en el momento de la ejecución de las pruebas de simulación, inicialmente se tomó un conjunto muy grande de valores (conjunto inicial) para uL y T , evaluando cada una de las posibles combinaciones y de conjuntos difusos, presento un dimensionamiento en el proceso de análisis y de solución, el cual, necesitaba más tiempo y más recursos en la máquina (computador /portátil), y en la herramienta de simulación. Recursos como capacidad de procesamiento y espacio para almacenar la información. Es por eso, que el algoritmo DMBAA-FLPSO se configura para determinar, en esta fase inicial, el conjunto difuso de cada parámetro de ensamble, uL y T , con tres valores, donde, una de las combinaciones posibles es la de menor probabilidad de bloqueo de ráfagas (PB_ráfagas) dentro del conjunto inicial.
- Para el desarrollo de este trabajo de grado de Maestría, se realizó un análisis exhaustivo para la elección del tiempo de OffSet durante el diseño de la red de prueba y ejecución de las pruebas de simulación, ya que este parámetro es un factor importante al optimizar los recursos disponibles y disminuir la pérdida de ráfagas. Además, para determinar el valor adecuado del tiempo de OffSet se debe tener en cuenta el tiempo que tarda el ensamblar la ráfaga, relacionado con el tamaño de la ráfaga a transmitir, es decir, con los parámetros de ensamble uL y T .



5.1. 3 Respecto a la herramienta de simulación

- OMNeT++ se considera como una elección acertada en cuanto a las características que brinda para la construcción de modelos de simulación. Sin embargo, es una herramienta con una curva de aprendizaje alta, ya que exige un conocimiento en programación orientada a objetos en C++ y requiere de un proceso de familiarización fuerte.
- Bajo las configuraciones establecidas en la red OBS Distribuida (red de prueba) la versión 4.6 de OMNeT++ para Windows, la cual fue utilizada para realizar el modelado de las redes, red de prueba con el algoritmo DMBA-FLPSO y la red de prueba con el algoritmo de ensamble Mixto Clásico, presenta problema de inestabilidad y bloqueo cuando se corre a grandes tiempos de simulación, con tiempos superiores a 1.5 segundos, lo cual, es una limitante cuando el parámetro variable es el tiempo. Debido a esto, en este trabajo de grado de Maestría, las simulaciones se realizaron con tiempos menores a 1.2s, con 0.2s para 0.1E y 0.2E (carga Muy Baja), con 0.4s para 0.3E y 0.4E (carga Baja), con 0.6s para 0.5E y 0.6E (carga Media), con 0.8s y 0.78s para 0.7E y 0.8E (carga Media) y con 0.78s y 1.2s para 0.9E y 1E (carga Muy Alta), ver tabla 4.12 y 4.14. De esta manera, no se tuvo ese inconveniente.
- La herramienta para la realización de las gráficas ORIGIN PRO 8.6 fue una buena elección, ya que esta presenta gran flexibilidad y facilidad de uso. Es una herramienta que permite diferentes formas de representar los datos según las necesidades del usuario y exportar las gráficas realizadas con alta calidad y a mejor escala. Donde, puede hacer diferentes ajustes de los datos de tipo lineal, polinómico, exponencial, etc. sobre la gráfica. Además, incluye herramientas para el análisis de datos y funciones como estadística avanzada, regresión y análisis de picos, entre otros.

5. 2 RECOMENDACIONES

- El IDE OMNeT++ es una herramienta robusta que consume gran cantidad de recursos, por lo que se recomienda el uso de equipos de cómputo de alta capacidad, en cuanto procesador (Intel Core i7 o AMD A10) y memoria RAM (8GB o más) para evitar problemas de bloqueo y exceso de tiempo en la simulación.
- Antes de ejecutar las simulaciones es importante adoptar y aplicar una metodología de simulación que permita realizar de manera organizada este proceso, esto con el objetivo de lograr menor complejidad, mayor claridad y una correcta implementación de los sistemas a simular.



- El uso de OMNeT presenta una complejidad elevada, por lo que se recomienda un entrenamiento previo en lenguaje de programación C++ con el fin de facilitar la implementación de los módulos.

5. 3 TRABAJOS FUTUROS

Con el desarrollo de este trabajo de grado de Maestría, surgen nuevas propuestas de investigación, tales como:

- Por la misma línea, en el proceso de ensamble de ráfagas de datos (problema), al analizar el desempeño en una red OBS Distribuida, los nuevos algoritmos de ensamble, partiendo de un algoritmo Mixto Híbrido y Adaptativo, deben enfocar la solución a mejorar el parámetro de desempeño el retardo extremo a extremo (e2e-D_ráfagas) junto con el retardo de ensamblaje, dado que, es una característica importante en las aplicaciones en tiempo real.
- Por la misma línea, en el proceso de ensamble de ráfagas de datos (problema), al analizar el desempeño en una red OBS Distribuida, los nuevos algoritmos de ensamble, partiendo de un algoritmo de ensamble Mixto Híbrido y Adaptativo, deben implementar el criterio de Diferenciación de Servicios por nivel de Congestión y por Predicción. Con el objetivo de hacer una comparación entre el algoritmo de ensamble (Mixto, Híbrido, Adaptativo, Diferenciación de Servicios) por nivel de Congestión y el mismo algoritmo de ensamble (Mixto, Híbrido, Adaptativo, Diferenciación de Servicios) pero por Predicción.
- Por la misma línea, en el proceso de ensamble de ráfagas de datos (problema), al analizar el desempeño en una red OBS Distribuida, los nuevos algoritmos de ensamble, partiendo de un algoritmo de ensamble Mixto Híbrido y Adaptativo, deben examinar el impacto de usar velocidades de transmisión de 5Gbps, 10Gbps o más, ya que en este trabajo se realizaron pruebas con una velocidad de 2.5Gbps.
- Analizar el desempeño de una red Óptica Flexible (FlexGrid) centralizada al implementar diferentes técnicas de asignación de espectro semi-elástico y elástico y compararla con una técnica de asignación fija. En mi caso, se puede incluir en este análisis, implementar el algoritmo de ensamble propuesto en este trabajo de grado de Maestría y comparar el desempeño de la red con un algoritmo de ensamble Mixto Clásico. Esto para implementar dos algoritmos en dos procesos dentro del esquema funcional de una red Óptica. Observación: esta idea, es la propuesta para mi doctorado.



REFERENCIAS

- [1] MINTIC. Boletín trimestral de las TIC. Cifras Primer Trimestre de 2018, Ministerio de Tecnologías de la Información y las Comunicaciones, Republica de Colombia. Variación porcentual de las conexiones Banda Ancha, cifras del 1T-2017 al 1T-2018. Disponible en:
http://colombiatic.mintic.gov.co/679/articles-75854_archivo_pdf.pdf.
- [2] MINTIC. Boletín trimestral de las TIC. Cifras Cuarto Trimestre de 2016, Ministerio de Tecnologías de la Información y las Comunicaciones, Republica de Colombia. Variación porcentual de las conexiones Banda Ancha, cifras del 4T-2015 al 4T-2016. Disponible en:
http://colombiatic.mintic.gov.co/602/articles-51235_archivo_pdf.pdf
- [3] UIT-T, ITU Telecommunication Standardization Sector, Spectral grids for WDM applications: CWDM frequency grid”, ITU-T Rec. G.694.1., June, 2002. Available: <https://www.itu.int/rec/T-REC-G.694.1/es>.
- [4] D. van den Borne, “Robust Optical Transmission Systems – Modulation and Equalization”, Doctoral Thesis, 2008
- [5] E. I. Mena y R. E. Mendoza, “Diseño de una red WDM para Andinatel S.A en la provincia de Tungurahua,” Tesis de Pregrado, Facultad de Ingeniería Electrónica, Escuela Politécnica Nacional, Quito, Ecuador, 2009.
- [6] CWDM/DWDM integration – an easy-to-realize expansion of existing CWDM systems (2012), Pan Dacom Direkt. [Online].
<https://www.pandacomdirekt.com/en/technologies/wdm/what-is-wdm.html>
[Last access: 8 October 2018].
- [7] I. Kaminov, T. Li, and A. Wilner, “Optical Fiber Telecommunications,” Vol. 5, Part A: Components and Subsystems, 5th Edition (Optics and Photonics), Elsevier, Amsterdam, Boston, London, Tokio, 2008.
- [8] M. Maier, “Optical Switching Networks”, Cambridge University Press, New York, USA, 2008.
- [9] J. Giovanni López, “DISEÑO DE MÉTODOS CROSS LAYER COGNITIVOS PARA REDES DE COMUNICACIÓN ÓPTICA DE RÁFAGAS (OBS)”. Tesis presentada a la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca para la obtención del Título de Doctor en Ciencias de la Electrónica, 2015.
- [10] R. J. Millan, “Estudio y Diseño de Multiplexores por División en Longitud de Onda (WDM) mediante efectos Electroópticos, Termoópticos y



- Acusoópticos,” Tesis Pregrado, Departamento de Teoría de la Señal y Comunicaciones, Universidad de Valladolid, Valladolid, España, 1998.
- [11] J. A. Meza, “CWDM TECNOLOGÍA Y APLICACIONES,” Trabajo de Grado de Pregrado, Facultad de Ingeniería Electrónica, Universidad de San Carlos de Guatemala, 2009.
- [12] F. Ramos, “Redes Ópticas basadas en CWDM,” Radiocomunicaciones y Fibra Óptica, radioptica.com. [Online]. Available: <https://archive.li/GfZml>.
- [13] A. Martin, “CONTRIBUCIÓN AL ESTUDIO Y OPTIMIZACIÓN DE DISPOSITIVOS BASADOS EN HOLOGRAFÍA DINÁMICA PARA SU USO EN REDES ÓPTICAS PASIVAS MULTIPLEXADAS EN LONGITUD DE ONDA WDM-PON,” Tesis Doctoral, Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicaciones, 2007.
- [14] A. J. Ordóñez, “Diseño y Simulación de los Efectos de FWM en una Red UDWDM PON,” Maestría en Redes de Comunicaciones, Facultad de Ingeniería, Quito, 2015.
- [15] UIT-T, ITU Telecommunication Standardization Sector, Spectral grids for WDM applications: DWDM frequency grid”, ITU-T Rec. G.694.2., December, 2003. Available: <https://www.itu.int/rec/T-REC-G.694.2/es>
- [16] S. Shekhar, “Survivable Multicasting in WDM Optical Networks,” Tesis Doctoral, Universidad de Pune, Maharashtra, India, Agosto de 2004.
- [17] Milos Kozak. “Efficient Control, Routing, and Wavelength Assignment in Loss-Less Optical Burst Switching Networks”. Czech Technical University in Prague, Dissertation Thesis PhD., 2015.
- [18] Natthapong Liamcharoen. “PERFORMANCE ANALAYSIS OF A PROPOSED HYBRID OPTICAL NETWORK”. University of Pittsburgh, PhD Candidate, 2016.
- [19] T. Venkatesh and C. Siva Ram Murthy. “Introduction to Optical Burst Switching”. In An Analytical Approach to Optical Burst Switched Network, New York USA: Springer Sciencie+Bussines Media, LLC, 2010, pp. 1-41.
- [20] J. Jue, and V. Vokkarane, “Optical Burst Switched Networks (Optical Networks)”. Libro. New York USA: Springer Sciencie+Bussines Media, Inc., 2005.
- [21] A. Lénin Almeida. “COMPORTAMIENTO DE REDES OBS (OPTICAL BURST SWITCHING) CON TRÁFICO TCP”. Escuela Politécnica Nacional, Proyecto Previo a la Obtención del Título de Ingeniero en Electrónica y Telecomunicaciones, Quito, 2010.



- [22] M. Klinkowski, F. Herrero, D. Careglio and J. Solé-Pareta. “Adaptive Routing Algorithms for Optical Packet Switching Networks”. In Proceedings of the 9th IFIP Working Conference on Optical Networks Design and Modelling, 2005.
- [23] Ch. Qiao, and M. Yoo. “Optical Burst Switching (OBS) – A New Paradigm for an Optical Internet”. University at Buffalo, New York, 1999, vol. 8, pp. 69-84.
- [24] M. Klinkowski, “Offset Time-Emulated Architecture for Optical Burst Switching – Modelling and Performance Evaluation”, Ph.D., dirigido por Dr. D. Careglio, Ph.D. Program: Arquitectura Tecnologia de Computadors, Universitat Politecnica de Catalunya, November. 2007.
- [25] Y. Chen, Ch. Qiao and X. Yu. “Optical Burst Switching (OBS): A New Area in Optical Networking Research”. *IEEE Network*, 2004, vol. 18, no. 3, pp. 16-23.
- [26] N. M. Garcia. “Architectures and Algorithms for IPv4/IPv6-Compliant Optical Burst Switching Networks”. Thesis submitted to the University of Beira Interior in candidature for the Degree of Doctor of Philosophy in Computer Science and Engineering, Covilha, Portugal, 2008.
- [27] M. Tena, “Evaluación de Arquitecturas de Red Híbridas OBS/OCS,” Tesis de Pregrado, Escuela Técnica Superior de Ingeniería de Telecomunicaciones de Barcelona, Barcelona, España, 2009.
- [28] S. Chagas, “*Uma Abordagem Distribuída para o Problema de Roteamento e Alocação de Comprimentos de Onda em Redes WDM*”, Tesis de doctorado, Universidad de Brasilia, Brasilia, Brasil, 2010.
- [29] Stepan Dumych. “Study on Traffic Aggregation Algorithms for Edge Nodes of Optical Burst Switching Network”. Lviv Polytechnic National University, Ukraine, 23-26 February, 2016, pp. 947-949.
- [30] A. Umaru A, M. Abd Latiff, and Y. Coulibaly. “A NOVEL FUZZY-BASED ADAPTIVE TIMER BURST ASSEMBLY ALGORITHM FOR OPTICAL BURST SWITCHING NETWORKS”. *Journal of Theoretical and Applied Information Technology*, September, 2014, vol. 67, no. 1.
- [31] A. Umaru, M. Abd Latiff, and Y. Coulibaly. “Fuzzy-Based Adaptive Hybrid Burst Assembly Technique for Optical Burst Switched Networks”. Hindawi Publishing Corporation, *Journal of Computer Networks and Communications*, Research Article, November, 2014, pp 10.



-
- [32] M. Mangwala and O. Ekabua. "A Survey of Burst Assembly Algorithms for Optical Burst Switching (OBS)". Research Article, International Journal of Engineering and Technology Research, Department of Computer Science, North West University, August, 2013, vol. 1, No. 7, pp. 107-115
- [33] A. Yayah, A. Samad and Y. Coulbaly. "A Review Burst Assembly Techniques in Optical Burst Switching (OBS)". Faculty of Computing, Universiti Teknologi Malaysia. Jurnal Teknologi, February, 2015, vol. 73, no. 2, Johor, Malaysia.
- [34] A. Yayah, A. Samad and Y. Coulbaly. "Burst Assembly Techniques in Optical Burst Switching (OBS)". Faculty of Computing, Universiti Teknologi Malaysia, 1st International Conference of Recent Trends in Information and Communication Technologies, Skudai, Johor, Malaysia, September, 2014
- [35] A. Gupta, R. Kaler, and H. Singh. "Investigation of OBS assembly technique based on various scheduling techniques for maximizing throughput". Journal Elsevier, Optik, 2013, vol. 124, pp. 840-844.
- [36] A. Kumar Garg. "A Novel Hybrid Approach for Efficient Network Utilization of OBS". International Journal of Software Engineering and its Applications. Department of Electronics and Communication Engineering. Mullana, India, January, 2012, Vol. 6, no 1.
- [37] Guoqiang Hu. "Quality of Service Guarantee in the Edge Node of Optical Packet/Burst Switched Networks with Traffic Assembly". Universitat Stuttgart zur Erlangung der Wurde eines Doktor-Ingenieurs (Dr. Ing.). Juli, 2009.
- [38] D. Patel, Dr. V. Dwivedi, and Dr. K. Bhatt. "A Novel Scheme for Efficient Bandwidth Utilization in Optical Burst Switched Network". International Journal of Research and Scientific Innovation (URSI), June, 2016, vol. 3.
- [39] V. Reena, P. Ramesh, and Dr. P. Nair. "Performance Evaluation of Efficient Assembly of TCP Traffic for Optical Burst Switching Networks". National Conference on System Design and Information Processing, 2013.
- [40] J. Prado, J. Bravo, y J. López. "Análisis del desempeño de una red OBS distribuida al integrar un método cognitivo basado en metaheurística para el ensamble de ráfagas". Universidad del Cauca, Trabajo de grado, pregrado, Grupo de I+D Nuevas Tecnologías en Telecomunicaciones –GNTT, Popayán, 2015.
- [41] X. Jiang, N. Zhu, and L. Yuan. "A Novel Burst Assembly Algorithm for OBS Networks Based on Burst Size and assembly Time Prediction". Department of Computer Science and Telecommunications Engineering. Journal of Computational Information Systems, Jiangsu University, 2013, vol. IX, no. 2, pp. 463-475.



-
- [42] A. Umaru, C. Yahaya, and M. Abd Latiff. "A Fuzzy-based Burst Assembly Approach to Reduce End-to-End Delay in OBS Networks". 5th International Conference on Photonics (ICP), Faculty of Computing, Universiti Teknologi Malaysia, Kuala Lumpur, Skudai, Malaysia, 2-4 September, 2014.
- [43] H-L Liu, and Sh. Jiang. "A mixed-length and time threshold burst assembly algorithm based on Traffic prediction in OBS network". International Journal of Computers Communications Control, 2012, vol. 2, pp. 87-93.
- [44] A. Sideri, and E. Vargarigos. "New Assembly Techniques for Optical Burst Switched Networks Based on Traffic Prediction". IFIP. Computer Engineering and Informatics Dept. International Federation for Information Processing, University of Patrás. Greece, 2007, pp. 358-367.
- [45] Viktors Zagorskis. "Using the Probabilistic Model Checker PRISM to analyze H-OTBA Algorithm in Optical Burst Switching (OBS) Networks". Advances in Wireless Communications (RTUWO). Telecommunications Institute, Riga, Latvia, 2015.
- [46] Z. Zhang, J. Luo, Q. Zeng, and Zhou. "Novel threshold-based burst assembly scheme for QoS support in optical burst switched WDM networks". Shanghai Jiaotong University, China. In Performance and Control of Next-Generation Communications Networks, August, 2003, pp. 250-256.
- [47] Z. Zhang, F. Cheng, J. Wang, J. Luo, Q. Zeng, and X. Xuan. "A new burst assembly and dropping scheme for service differentiation in optical burst switched networks". In Proceedings of the Asia-Pacific Optical and Wireless Communications: Optical Transmission. Switching and Subsystems (APOC '03). Shanghai Jiaotong Univ., Wuhan, China, May, 2004, pp. 236-245.
- [48] V. Vokkarane, K. Haridoss, and J. Jue. "Threshold-based burst assembly policies for QoS support in optical burst-switched networks". Center for Advanced Telecommunications Systems and Services. The University of Texas at Dallas. In Proceedings of the Optical Networking and Communications (OptiComn '02), Boston, USA, July, 2002, vol. 4874, pp. 125-136.
- [49] K. Dolzer, and C. Gauger. "On Burst Assembly in Optical Burst Switching Networks –A Performance Evaluation of Just-Enough-Time". 17th International Teletraffic Congress, University of Stuttgart, Institute of Communication Networks and Computer Engineering, Salvador, Brazil, September, 2001, vol. 4, pp. 149-160.
- [50] V. Kavitha, and V. Palanisamy. "NEW BURST ASSEMBLY AND SCHEDULING TECHNIQUE FOR OPTICAL BURST SWITCHING



-
- NETWORKS". Journal of Computer Science, Department of Electronics and Communication Engineering, 2013.
- [51] M. Duser, and P. Bayvel. "Performance of a Dynamically Wavelength-Routed Optical burst switched network". IEEE Photonics Technology Letters, February, 2002, vol. 14, no 2, pp. 239-241.
- [52] I. Karamitsos and Ch. Bowerman. "A Resource Reservation Protocol with Linear Traffic Prediction for OBS Networks". Research Article, Hindawi, Hindawi Publishing Corporation, Advances in Optical Technologies, Department of Computing, School of Engineering, University of Sunderland, 2013, vol. 2013, pp. 132-148.
- [53] A. GE, F. Callegati, and L. Tamil. "On Optical Burst Switching and Self-Similar Traffic". IEEE Communications Letters, March, 2000, vol. 4, no. 3, pp. 98-100.
- [54] X. Cao, J. Li, Y. Chen, and Ch. Qiao. "Assembling TCP/IP Packets in Optical Burst Switched Networks". Department of Computer Science and Engineering. State University of New York at Buffalo. In Proceedings of the IEEE Global Telecommunications Conference, (GLOBECOM '02), November, 2002, pp. 17-21, Taipei, Taiwan.
- [55] G. Hu, K. Dolzer, and Ch. Gauger. "Does burst assembly really reduce the self-similarity.?" In Optical Fiber Communication Conference. Optical Society of America, University of Stuttgart, Institute of Communication Networks and Computer Engineering, 2003.
- [56] B. Kantarci, and S. Oktug. "ADAPTIVE THRESHOLD BASED BURST ASSEMBLY IN OBS NETWORKS". In Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE '06), Ottawa, Canada, May, 2006, pp. 1419-1422.
- [57] A. Yayah, Y. Coulibaly, A. Ismail, and G. Rouskas. "Hybrid offset-time and burst assembly algorithm (H-OTBA) for delay sensitive applications over optical burst switching networks". International Journal of Communication Systems, Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia, May, 2014, vol. 29, no. 2, pp. 251-261.
- [58] M. Dawood, M. Mahmoud, and M. H.Aly. "Adaptive Data Burst Assembly in OBS Networks". International Journal of Research and Scientific Innovation (URSI), June, 2016, vol. 3.
- [59] V. Tavanam, D. Venkateswarlu, and K. Dasari. "BFCA-VF: Best Fit Channel Allocation and Void Filling by Burst Segmenting and Scheduling". IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), India, May, 2014.



-
- [60] R. Adgaonkar, and S. Sharma. "A Review of Burst Scheduling Algorithm in WDM Optical Burst Switching Network". IJCSI International Journal of Computer Science Issues, November, 2011, vol. 8, no 3.
- [61] A. Kumar Garg. "Traffic prediction based burst assembly mechanism for OBS". Department of Electronics & Communication Engg. Deenbandhu Chhotu Ram University of Science & Technology, India, August, 2013, vol 124, pp. 2017-2019.
- [62] K. Seklou, A. Sideri, P. Kokkinos, and E. Varvarigos. "New assembly techniques and fast reservation protocols for optical burst switched networks based on traffic prediction". Journal Optical Switching and Networking, Elsevier, Department of Computer Engineering and Informatics, University of Patras and Research Academic Computer Technology Institute, Patras, Greece, 2013, pp. 132-148.
- [63] X. Yu, Y. Chen, and Ch. Qiao. "A Study of traffic statistics of assembled Burst traffic in Optical Burst switched networks". Department of Computer Science and Engineering, State University of New York at Buffalo, 2002, pp. 149-159.
- [64] R. Neapolitan, and K. Naimipour, "Foundations of Algorithms," Jones and Bartlett Publishers, 1997.
- [65] E. Horowitz, S. Sahni, and S. Rajasekaran, "Computer Algorithms," Computer Science Press, 1998.
- [66] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," The MIT Press/McGraw-Hill, 2001.
- [67] A. Cruz, "ALGORITMOS DE APROXIMACIÓN DISTRIBUIDOS PARA LA DISEMINACIÓN DE INFORMACIÓN EN MANETS 3D," Tesis presentada para obtener el grado de Maestría en Ciencias de la Computación, dirigida por Dr. R. Menchaca, and Dr. F. J. Zaragoza, Instituto Politécnico Nacional, 2012.
- [68] C. E. Maldonado, "HEURÍSTICA Y PRODUCCIÓN DE CONOCIMIENTO NUEVO EN LA PERSPECTIVA CTS," Estética, Ciencia y Tecnología. Creaciones electrónicas y numéricas, Bogotá, Editorial Pontificia Universidad Javeriana, pp. 98-127, June. 2014.
- [69] C. J. Sarmiento, "MÉTODO PARTICLE SWARM OPTIMIZATION (PSO – ENJAMBRE DE PARTÍCULAS) APLICADO AL PROBLEMA DE MÚLTIPLES OBJETIVOS DEL JOB SHOP SCHEDULING (JSP) O SECUENCIAMIENTO DE MÁQUINAS," Presentado como Trabajo de Grado



- para optar el título de Ingeniera Industrial, dirigido por Ph.D H. Lamos, Bucaramanga, Mayo. 2012.
- [70] F. Luna, “Metaheurísticas avanzadas para problemas reales en redes de telecomunicaciones,” Tesis Doctoral, dirigido por Dr. E. A. Torres, and Dr. A. J. Nebro, Universidad del Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Abril. 2008.
- [71] S. Luke, “Essentials of Metaheurística,” Department of Computer Science, George Mason University, Second Edition, Online Version 2.1, October. 2014.
- [72] A. Vidal, “ALGORITMOS HEURÍSTICOS EN OPTIMIZACIÓN,” dirigido por M. L. Carpena, B. Casas, Máster en Técnicas Estadísticas, Universidad de Santiago de Compostela, Julio. 2013.
- [73] D. Morillo, L. Moreno, and J. Díaz, “Metodologías Analíticas y Heurísticas para la solución del Problema de Programación de Tareas con Recursos Restringidos (RCPSP): una revisión. Parte 1,” Ingeniería y Ciencia, vol. 10, no. 19, pp. 247-271, Enero. 2014.
- [74] J. F. Chicano, “Metaheurísticas e Ingeniería del Software,” Tesis Doctoral, dirigido por Dr. E. A. Torres, Universidad del Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Julio. 2007.
- [75] M. Gómez, “SISTEMA DE GENERACIÓN ELÉCTRICA CON PILA DE COMBUSTIBLE DE ÓXIDO SOLIDO ALIMENTADO CON RESIDUOS FORESTALES Y SU OPTIMIZACIÓN MEDIANTE ALGORITMOS BASADOS EN NUBES DE PARTICULAS,” Tesis Doctoral, dirigido por Dr. D. F. Jurado, and Dr. D. J. Carpio, Universidad Nacional de Educación a Distancia, Escuela Técnica Superior de Ingenieros Industriales, Capitulo 3, 2008. <http://e-spacio.uned.es/fez/view/tesisuned:IngInd-Mgomez>, http://e-spacio.uned.es/fez/eserv/tesisuned:IngInd-Mgomez/TESIS_MGG1_Capitulo3.pdf.
- [76] C. A. Correa, G. A. Camacho, R. A. Bolaños, and M. Granada, “Análisis comparativo de técnicas metaheurísticas para resolver el planeamiento de la transmisión,” pp. 81-105, Mayo. 2014.
- [77] A. Hinojosa, K. Espinosa, and R. Correa, “EL MÉTODO DE ENJAMBRE DE PARTÍCULAS Y EL CRITERIO DE MÍNIMA ENTROPÍA EN EL DISEÑO ÓPTIMO DE UN DISIPADOR DE CALOR,” Revista Ingenierías Universidad de Medellín –RIUM, vol. 11, no. 20, pp. 203-214, Mayo. 2012.



-
- [78] M. A. Muñoz, J. A. López, and E. F. Caicedo, "Inteligencia de enjambre: sociedades para la solución de problemas (una revisión)," *Revista Ingeniería e Investigación*, vol. 28, no. 2, pp. 119-130, Agosto. 2008.
- [79] H. Lamos, S. A. Galvan, L. J. Gonzalez, and C. C Jiménez, "Algoritmo PSO-Híbrido para solucionar el problema de ruteo de vehículos con entrega y recolección simultáneas," *Revista Facultad de Ingeniería –UPTC*, vol. 22, no. 35, pp. 75-90, Noviembre. 2013.
- [80] J. D. Prado, J. C. Bravo, and J. G. López, "HMCNA AS A New Algorithm for Burst Assembly on an OBS Distributed Network," *SAI Computing Conference*, London, UK, pp. 906-912, July. 2016.
- [81] C. González, "Lógica Difusa, una introducción práctica", *Técnicas de Softcomputing*, Libro. Universidad de Castilla-La Mancha, 2011.
http://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf
- [82] H. N. Acosta, and J. G. Salas, "DISEÑO DE CONTROLADORES DEDICADOS A LA LÓGICA DIFUSA", Tesis Doctoral, dirigido por J. Garrido, Universidad Autónoma de Madrid, Escuela Politécnica Superior, Madrid, España, Octubre. 2006.
- [83] V. Vega, "CONTROLADOR NEURONAL DIFUSO: ANALISIS DE ESTABILIDAD", Tesis Doctoral, Universidad Autónoma de Nuevo Leon, Facultad de Ingeniería Mecánica y Eléctrica, Programa Doctoral en Ingeniería Eléctrica, Monterrey, Marzo. 1995.
- [84] D. F. Matía, "ANÁLISIS Y DISEÑO DE SISTEMAS DE CONTROL DE PROCESOS BASADOS EN LÓGICA BORROSA", Tesis Doctoral, dirigido por D. A. Jiménez, and D. R. Galán, Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, 1994.
- [85] J. A. Delgado, "LOGICA DIFUSA APLICADA A LAS TELECOMUNICACIONES DE DATOS", Tesis Doctoral, Universidad Autónoma de Nuevo Leon, Facultad de Ciencias Fisico-Matematicas, División de Estudios de Posgrados, San Nicolas de los Garza, Septiembre. 1999.
- [86] R.J. Navas, "DISEÑO Y APLICACIÓN DE CONTROLADORES NEURO-DIFUSOS INTEGRADOS EN MODO MIXTO DE ALTA COMPLEJIDAD", Tesis Doctoral, dirigida por F. Vidal, Departamento de Electrónica, Universidad de Málaga, Diciembre. 2000.
- [87] M. Rojas, D. López, y C. Rincón, "APLICACIÓN DEL MÉTODO CYBER SIN CON LÓGICA DIFUSA EN UN SISTEMA DE CONTROL DE GESTIÓN,"



-
- Revista Ingenierías Universidad de Medellín –RIUM, vol. 11, no. 20, pp. 153-164, Mayo. 2012.
- [88] Ch. Chien, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller –Part I,” vol. 20, no. 2, Abril. 1990.
- [89] P. Tremante, and E. Brea, “Una visión de la teoría difusa y los sistemas difusos enfocados al control difuso,” Revista Ingeniería Industrial. Actualidad y Nuevas Tendencias, Ingeniería Industrial, Universidad de Carabobo, Carabobo, Venezuela, pp. 121-136, vol. 4, no. 12, 2014.
- [90] O. G. Duarte, “Sistemas de lógica difusa. Fundamentos,” Departamento de Ingeniería Eléctrica, Universidad Nacional de Colombia, Revista Ingeniería e Investigación, no. 42, pp. 22-30, Abril. 99.
- [91] D. Guzmán, and V. M. Castaño, “LA LÓGICA DIFUSA EN INGENIERÍA: PRINCIPIOS, APLICACIONES Y FUTURO,” Universidad Nacional Autónoma de México, Campus Juriquilla, Querétaro, México, pp. 87-107. 2006
- [92] A. F. Camargo, and E. J. Villamizar, “CONTROL DE TEMPERATURA DE UN HORNO POR MEDIO DE LOGICA DIFUSA”, Universidad Pontificia Bolivariana, Escuela de Ingeniería y Administración, Facultad de Ingeniería Electrónica, Bucaramanga, 2008.
- [93] K. Guevara y Y. Díaz, “Impacto de los mecanismos de control de contención en redes OBS distribuidas”, Trabajo de Grado de Pregrado, Universidad del Cauca, Popayán, Colombia, 2015.
- [94] A. Vargas, OpenSim Ltd, “OMNeT++ user’s manual”, OMNeT++ versión 5.4.1, 2016.
- [95] R. S. Pressman, “Ingeniería de Software un Enfoque Práctico”, Séptima Edición, Editorial McGraw-Hill, ISBN 6071503140, Madrid, 2010.
- [96] C. Hernández, y J. Vicente, “Introducción al simulador de redes NS-2”, Universidad Politécnica de Valencia, Valencia, España, 2011. Disponible en:
<https://riunet.upv.es/bitstream/handle/10251/12735/Art%C3%ADculo%20do%20cente%20NS-2.pdf?sequence=1>
- [97] M. Morales, M. Calle, J. Tovar, y J. Cuellar, “Introducción-Simulando con OMNeT: selección de la herramienta y su utilización”, Universidad ICESI, Cali, Colombia, 2013. Disponible en:
http://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/68446/1/des%20simulando_omnet_capitulouno.pdf



-
- [98] Departamento de Ingeniería Telemática, “OPNET: Manual de Usuario”, Universidad Politécnica de Cataluña, Barcelona, España, 2004-2005. Disponible en:
http://ansat.es/soporte/docs/fragmentacion/OPNET_Modeler_Manual.pdf
- [99] M. Espinosa, “Modelado de Mecanismos de Transición a IPv6”, Tesis Licenciatura, Ingeniería en Electrónica y Comunicaciones, Escuela de Ingeniería y Ciencias, Universidad de las Américas Puebla, México, 2007.
- [100] OMNeT++ User manual Version 4.6, OMNeT, 2014. Disponible en:
<https://omnetpp.org/doc/omnetpp4/Manual.pdf>
- [101] MathWorks, Trabajo con objetos gráficos y Análisis de datos. Disponible en:
https://es.mathworks.com/help/matlab/learn_matlab/understanding-handle-graphics-objects.html
https://la.mathworks.com/help/matlab/learn_matlab/data-analysis.html
- [102] Origin 2018, gráficas de alta calidad para diferentes tipos de grafos, características principales según la versión. Disponible en:
<https://www.originlab.com/index.aspx?go=Products/Origin&PID=1029>
- [103] D. Pinto, and B. Barán, “Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach,” 2nd IFIP/ACM Latin American Networking Conference 2005 – LANC’05. Cali, Colombia, Octubre. 2005.
- [104] UIT-T, Sector de Normalización de las Telecomunicaciones de la UIT, “Secciones digitales y sistemas digitales de línea –Sistemas de línea óptica para redes de acceso y redes locales”, ITU-T Rec. G.984.1., 2003. Available: <https://www.itu.int/rec/T-REC-G.984.1/es>.



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA



ANEXO A

Ing. Neil Andrés Jiménez Pinzón

Director: PhD. Jose Giovanni López Perafán

Universidad del Cauca

Facultad De Ingeniería Electrónica y Telecomunicaciones
Maestría en Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Grupo I+D Nuevas Tecnologías en Telecomunicaciones – GNTT
Popayán, Junio de 2019



ANEXO A. ALGORITMOS

A.1 Algoritmo para la generación y envío de paquetes.

Se generan paquetes por intervalos de tiempo, a los cuales se les asigna atributos como su dirección origen, dirección de destino y longitud. La dirección de destino se obtiene aleatoriamente mediante una función uniforme, encargada de escoger un número entre cero y el número de nodos de la red

```
if (msg == generatePacket)
{
    // Manejo y envío del paquete

    int destAddress = destAddresses[intuniform(0, destAddresses.size()-1)];
    //int destAddress = 1;

    char pkname[40];

    sprintf(pkname,"to-%d-#%ld", destAddress, pkCounter++);

    EV << "generating packet " << pkname << endl;

    Packet *pkt = new Packet(pkname);

    pkt->setByteLength(packetLengthBytes->longValue());

    pkt->setSrcAddr(myAddress);

    pkt->setDestAddr(destAddress);

    // Se establece la prioridad

    int type = intuniform(0,2);

    pkt->setPriority(type);

    send(pkt,"out");

    scheduleAt(simTime() + sendIATime->doubleValue(), generatePacket);

    if (ev.isGUI()) bubble("Generating packet..");
}
```




A.2 Algoritmo para la obtención de reglas por dirección de destino

Este algoritmo se encarga de obtener el archivo correspondiente a las reglas y comprara cada línea usando la clase "**Classifier_Rules**", la cual verifica si la dirección del paquete son compatibles con las reglas consignadas en el archivo.

```
if (numOuts != 0){  
    //Almacena la información proveniente de Clasifier_Rules  
    rules = (Classifier_Rules*)calloc(numOuts,sizeof(Classifier_Rules)); //se usa calloc  
    para reservar memoria, de forma dinámica  
    //lee una línea cada vez y crea la regla asociada para cada cola.  
    char *line = (char*)calloc(1500,sizeof(char)); //Debido a esto, el máximo número  
    de caracteres por línea es 1500.  
    int i=0;  
    //abre el archivo Rules.dat y lo almacena en rulesFile  
    const char *rulesFile = par("rules");  
    //Si rules está vacío, se muestra un mensaje de error y para la simulación.  
    if(strlen(rulesFile) == 0){  
        opp_error("El archivo de reglas no se encuentra definido");  
    }  
  
    // abre el archivo Rules.Dat en *ruleFile  
    FILE *ruleFile = fopen(rulesFile,"r"); // r = leer el archivo... w=escribir...d=Borrar  
    if(ruleFile != NULL){  
        //fgets lee hasta que se encuentra un carácter \n  
        while(fgets(line,1500,ruleFile) != NULL){  
            if(strcmp(line,"\n") != 0 && line[0] != '#'){ //Ignora los comentarios (líneas que  
            empiezan con #)  
                //toma "line" como parámetro de entrada  
                //para realizar la comparación de los parámetros
```



```
rules[i] = Classifier_Rules(line);

i++;

} } }

else{

opp_error("No es posible abrir el archivo de reglas");

}

fclose(ruleFile);

// i debe ser igual a numQueues. Si no lo es, seguramente hay un error.

if(!(i == numOuts)){

printf("(Classifier_Rules) Aviso: El despachador de reglas no coincide con las
colas de los modulos.\n");

}

free(line);

} }
```

A.3 Algoritmo para establecer el criterio de ensamble

Este algoritmo recibe los paquetes y los almacena en una cola mediante tres criterios: número máximo de paquetes, el tamaño en Bytes y el tiempo de ensamblaje. Cuando alguno de los tres criterios se cumpla, se ensambla la ráfaga con el método “assemblyBurst”.

// Comprobar si el busrtifier está vacío

```
if(burst.empty()) // Si lo está, inicializar el tiempo máximo y la cola

{

scheduleAt(simTime() + maxTime, maxTime_msg ); //Establece el tiempo
máximo

//Registrar el momento en el que llega el primer paquete

firstPacket_t = simTime();

}
```



```
else if(desborde && !addLastPacket)

    { // Entra si se necesita ensamblar la ráfaga antes de que el paquete sea
    insertado en la cola

        if(burst.empty())opp_error("No es posible ensamblar una ráfaga con una cola
vacía");

        //Ensamblar la ráfaga e iniciar contadores

        assemblyBurst();

        if(maxTime_msg->isScheduled())cancelEvent(maxTime_msg); //Se cancela el
tiempo maximo y se programa uno nuevo

        scheduleAt(simTime() + maxTime, maxTime_msg);

        firstPacket_t = simTime();

        //Calcular si la sobrecarga se da con el primer paquete

        desborde = false;

        if(((burstBits + pqt->getBitLength()+ tamHeaderPacketBits)> maxSizeBits))
desborde=true;

    }

    //Insertar el paquete actual en la cola

    timeAssembled.collect(simTime()-firstPacket_t);

    burst.insert(pqt);

    burstBits += pqt->getBitLength() + tamHeaderPacketBits;

    numPacketsInBurst++;

    //Si la sobrecarga no está habilitada, pero ocurre sobrecarga cuando se inserta el
primer paquete, se genera un error.

    if(desborde && !addLastPacket)

        opp_error("La sobrecarga se generó insertando el primer mensaje y los
requerimientos no permiten sobrecarga(overflowLastPacket es falso)");

        //Si ocurre la sobrecarga o se llena al máximo número de paquetes se debe
ensamblar la ráfaga.

        if((desborde || numPacketsInBurst == numPackets) || burstBits ==maxSizeBits)

            {
```



```
assemblyBurst();  
  
if(maxTime_msg->isScheduled()) cancelEvent(maxTime_msg);  
  
} }
```

A.4 Algoritmo para programar el envío del BCP

Este algoritmo verifica que la ráfaga haya llegado desde el modulo “**Burstifier**”, se encarga de encontrar el canal mediante el horizonte más cercano con la función “**findNearestHorizon**”.

```
if(!(msg->isSelfMessage())) // Recibe una ráfaga desde BurstAssembler (Burstifier)  
{  
    //Emite el mensaje a ráfaga  
    Burst *rfg = check_and_cast<Burst*>(msg);  
    burstTam.record(rfg->getBitLength());  
    burstRecv++;  
    int wl = 0;  
    int pos=0;  
  
    wl=findNearestHorizon(); //retorna la posición del horizonte más apto para enviar la  
    ráfaga actual  
  
    char pfname[40];  
    sprintf(pfname,"----Wavelength--%d",wl);  
  
    EV << "info" << pfname << endl;  
  
    //Inserta el id de la ráfaga  
    rfg->setSenderId(getId()); //retorna el identificador del módulo  
  
    BurstSenderInfo *myInfo = new BurstSenderInfo();  
  
    myInfo->setIdBurstifier(rfg->getIdBurstifier()); //retorna el índice del vector  
    correspondiente al módulo  
  
    myInfo->setNumSeq(rfg->getSecNum()); //Contador de  
    ráfagas...burstCounter este sec num sirve cuando los paquetes o ráfagas llegan  
    desordenados, con este indicador, se puede ordenar cuando lleguen al receptor
```



```
myInfo->setAssignedLambda(wl);

BurstifierInfo *lInfo = (BurstifierInfo*) rfg->removeControllInfo();

myInfo->setLabel(lInfo->getLabel()); //se define en el .ini una etiqueta de destino
para cada burstifier

delete lInfo;

//Crea el automensaje "schedule the bcp_ini send"

cMessage *ctrMsg= new cMessage("Sched");

//Entra aquí si la ráfaga puede ser enviada solo cuando el canal wl se establezca
como libre. (habrá mucho tiempo para enviar el bcp y esperar el tiempo de offset
máximo)

if(horizon[wl] - rfg->getMaxOffset() >= simTime()) //el = Significa que se puede
programar una ráfaga al mismo tiempo que el valor del horizonte

{

//Se almacena en ScheduledBurst

pos = scheduleBursts.insertBurst(rfg, horizon[wl]);

if(pos == -1)

{ //scheduledBurst está lleno. Se decarta esta ráfaga

delete msg;

delete ctrMsg;

burstDroppedByQueue++;

return;

}

//Inserta la posición de la ráfaga en la cola de scheduledBurst

myInfo->setBurstId(pos);

//Se llenan los campos de control(esteste mensaje viajará a través de todos los
estados del sender)

ctrMsg->setControllInfo(myInfo);

ctrMsg->setKind(OBS_PROGRAMAR_BCP); //Se Establece en el paso 1: Schedule
BCP

//Programar envío de BCP

scheduleAt(horizon[wl] - rfg->getMaxOffset(), ctrMsg);
```




```
//Actualizar el valor del horizonte

horizon[wl] = horizon[wl] + (rfg->getBitLength()/dataRate) + guardTime;

//Se Registra el valor del horizonte
horizonVec[wl]->record(horizon[wl]);
}

else //Se puede mandar el BCP inmediatamente de esta forma la ráfaga será
enviada dentro del offset máximo
{
//Se almacena en ScheduledBurst
pos = scheduleBursts.insertBurst(rfg, simTime() + rfg->getMaxOffset());
char name[40];
sprintf(name, "--pos %d", pos);
EV<< "info"<<name<< endl;
if(pos == -1)
{ // la cola de ScheduleBursts queue está llena. Descartar
delete msg;
delete ctrMsg;
burstDroppedByQueue++;
return;
}
//Insertar la posición de la ráfaga en la cola scheduledBurst
myInfo->setBurstId(pos);

// LLenar los campos del mensaje de control(Este mensaje pasará a traves de
todos los estados del sender)
ctrMsg->setControlInfo(myInfo);
ctrMsg->setKind(OBS_PROGRAMAR_BCP);
//Programar envío de BCP en este momento
scheduleAt(simTime(), ctrMsg);

//Actualizar el valor del horizonte
```



```
horizon[wl] = simTime() + rfg->getMaxOffset() + (rfg->getBitLength()/dataRate) +  
guardTime;  
  
//Registrar el valor del horizonte  
  
horizonVec[wl]->record(horizon[wl]);  
  
} }
```

A.5 Algoritmo para el desensamble de la ráfaga

```
if(dynamic_cast< Burst *> (msg) == NULL){ delete msg; return; } //El paquete de  
control de la ráfaga no debería pasar en este punto
```

```
if(msg->getKind() == 1){ //Inicio de la ráfaga. Coloca este mensaje en la cola
```

```
take(msg); //toma posesión
```

```
Burst *recvBurst = check_and_cast < Burst*> (msg);
```

```
receivedBursts.push_back(recvBurst);
```

```
//Adiciona la ráfaga en el contador de ráfagas recibidas
```

```
recvBursts++;
```

```
listSize++;
```

```
VlistSize.record(listSize);
```

```
}
```

```
else if(msg->getKind() == 2){ //Final de la ráfaga. Se busca el inicio del mensaje y se  
desensambla.
```

```
Burst *recvBurst = check_and_cast < Burst*> (msg);
```

```
int bld,nSeq; // valores de ID de la ráfaga
```

```
bld = recvBurst->getIdburstifier();
```

```
nSeq = recvBurst->getSecNum();
```

```
list<Burst*>::iterator i;
```

```
Burst* actElem;
```

```
//Busca desde el comienzo porque la ráfaga que se está buscando es  
probablemente una de la últimas
```

```
for(i = receivedBursts.begin(); i != receivedBursts.end(); i++){
```



```
actElem = *i;

if((actElem->getIdBurstifier() == bld) && (actElem->getSecNum() == nSeq))
break; //Burst found!

}

if(i != receivedBursts.end()){ //Si el iterador superior no llega hasta el final...significa
que la ráfaga fue encontrada.

    Burst *burstIni= check_and_cast< Burst* > (*i);

    cMessage *tempPack;

    while(burstIni->hasMessages()){ //Libera los paquetes hasta que la cola de
burstIni queue esté vacía

        tempPack = burstIni->retrieveMessage();

        //TODO: Envía el paquete a un buffer intermedio por lo tanto, todos los
paquetes no serían liberados a la red eléctrica al mismo tiempo.

        send(tempPack,"out");

    }

    //Limpieza

    delete msg;

    delete burstIni;

    i = receivedBursts.erase(i);

    listSize--;

    VlistSize.record(listSize);

}

else{

    printf("<OBS_BurstDissassembler><t=%s> Error!! burst with id=(%d,%d) not
found!\n",simTime().str().c_str(),bld,nSeq);

    delete msg;

} }

}
```



ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS
EN UNA RED OBS DISTRIBUIDA

ALGORITMO ADAPTATIVO BASADO EN LÓGICA DIFUSA Y PSO PARA EL ENSAMBLE DE RÁFAGAS EN UNA RED OBS DISTRIBUIDA



ANEXO B

Ing. Neil Andrés Jiménez Pinzón

Director: PhD. Jose Giovanni López Perafán

Universidad del Cauca

Facultad De Ingeniería Electrónica y Telecomunicaciones
Maestría en Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Grupo I+D Nuevas Tecnologías en Telecomunicaciones – GNTT
Popayán, Junio de 2019



ANEXO B. RESULTADOS DE SIMULACIÓN PARA 5Gbps

B.1 Resultados obtenidos del porcentaje de probabilidad de bloqueo (% PB), en función del tráfico de entrada variable a una red OBS distribuida NSFNET, para 5.0 Gbps por Lambda, con 2 Lambdas (para carga útil –ráfagas) por Enlace de Fibra (DWDM).

Tabla B.1 Porcentaje de probabilidad de bloqueo (% PB) vs Tráfico (variable) para 5.0 Gbps por Lambda.

Tráfico (E)	% Probabilidad de Bloqueo (% PB) Para 5.0 Gbps por λ . Con 2 λ por enlace.	
	Algoritmo Mixto (Clásico)	Algoritmo DMBA-FLPSO (Propuesto)
1	22,5614010740406	11,2002805408214
0,9	22,3420122050824	11,0014594254557
0,8	21,5437689614494	10,4743214236918
0,7	21,1048780485989	10,1854536156278
0,6	19,4024155633810	10,4919852439484
0,5	19,1286734573951	10,1976207610321
0,4	17,7918444686290	10,7890170707157
0,3	17,5059310860920	10,5890699214791
0,2	16,5096450043329	10,0012763330271
0,1	16,5096450043329	9,3471916135560

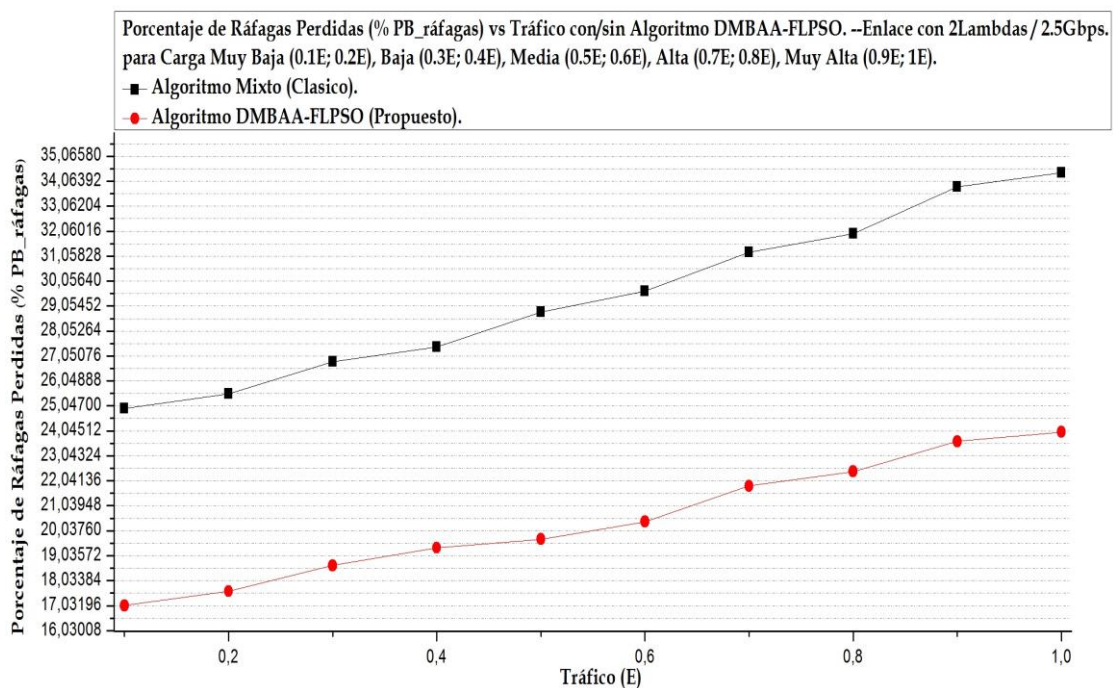


Figura B.1 Porcentaje de probabilidad de bloqueo (%PB) vs tráfico (variable) para 5.0 Gbps por λ .



B.2 Resultados obtenidos del porcentaje del retardo extremo a extremo (% e2e-D), en función del tráfico de entrada variable a una red OBS distribuida NSFNet, para 5.0 Gbps por Lambda, con 2 Lambdas (para carga útil –ráfagas) por Enlace de Fibra (DWDM).

Tabla B.2 *Porcentaje del retardo extremo a extremo (% e2e-D) vs Tráfico (variable) para 5.0 Gbps por Lambda.*

Tráfico (E)	% Retardo extremo a extremo (% e2e-D) Para 5.0 Gbps por λ . Con 2 λ por enlace.	
	Algoritmo Mixto (Clásico)	Algoritmo DMBAA-FLPSO (Propuesto)
1	0,40513288010913	0,405166493524207
0,9	0,40513219802361	0,405166010686492
0,8	0,40513022348238	0,405162810795124
0,7	0,40512950206776	0,405159074122462
0,6	0,40512818320663	0,405146714973804
0,5	0,40512743309023	0,405145498038630
0,4	0,40512642379294	0,405137297808158
0,3	0,40512607880431	0,405136330540948
0,2	0,40512481153725	0,405134745645269
0,1	0,40512443515570	0,405133401330078

Porcentaje del Retardo extremo a extremo (% e2e-D_ráfagas) vs Tráfico con/sin Algoritmo DMBAA-FLPSO. --Enlace con 2_Lambdas /5.0_Gbps. para Carga Muy Baja (0.1E; 0.2E), Baja (0.3E; 0.4E), Media (0.5E; 0.6E), Alta (0.7E; 0.8E), Muy Alta (0.9E; 1E).
 ■ Algoritmo Mixto (Clásico).
 ● Algoritmo DMBAA-FLPSO (Propuesto).

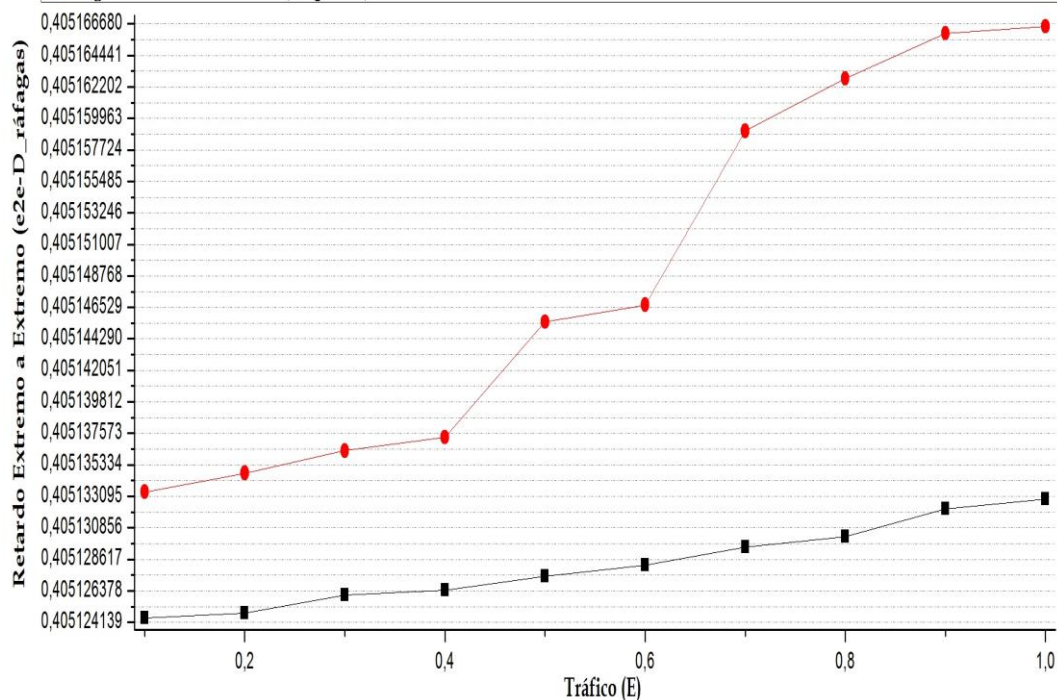


Figura B.2 *Porcentaje del retardo extremo a extremo (% e2e-D) vs tráfico (variable) para 5.0 Gbps por λ .*



B.3 Resultados obtenidos del porcentaje de probabilidad de bloqueo (% PB), en función del tráfico de entrada variable a una red OBS distribuida NSFNet, para 2.5 Gbps y 5.0 Gbps por Lambda, con 2 Lambdas (para carga útil –ráfagas) por Enlace de Fibra (DWDM).

Tabla B.3 Porcentaje de probabilidad de bloqueo (% PB) vs Tráfico (variable) para 2.5 Gbps y 5.0 Gbps por Lambda.

E	%Probabilidad de Bloqueo (%PB) Para 2.5 Gbps por λ . Con 2 λ por enlace.		%Probabilidad de Bloqueo (%PB) Para 5.0 Gbps por λ . Con 2 λ por enlace.	
	Algoritmo Mixto (Clásico)	Algoritmo DMBAА-FLPSO (Propuesto)	Algoritmo Mixto (Clásico)	Algoritmo DMBAА-FLPSO (Propuesto)
1	34,7010587500	24,2241401305	22,5614010740406	11,2002805408214
0,9	33,7311946019	23,4538255289	22,3420122050824	11,0014594254557
0,8	31,9691531124	21,9703518713	21,5437689614494	10,4743214236918
0,7	31,2010901257	21,3338670838	21,1048780485989	10,1854536156278
0,6	29,6536835368	20,4004146088	19,4024155633810	10,4919852439484
0,5	28,8121313406	19,6975381256	19,1286734573951	10,1976207610321
0,4	27,4112463274	19,3511663859	17,7918444686290	10,7890170707157
0,3	26,8225749497	18,6414195915	17,5059310860920	10,5890699214791
0,2	25,5329195722	17,6125402701	16,5096450043329	10,0012763330271
0,1	24,9330153764	17,0375620451	16,5096450043329	9,3471916135560

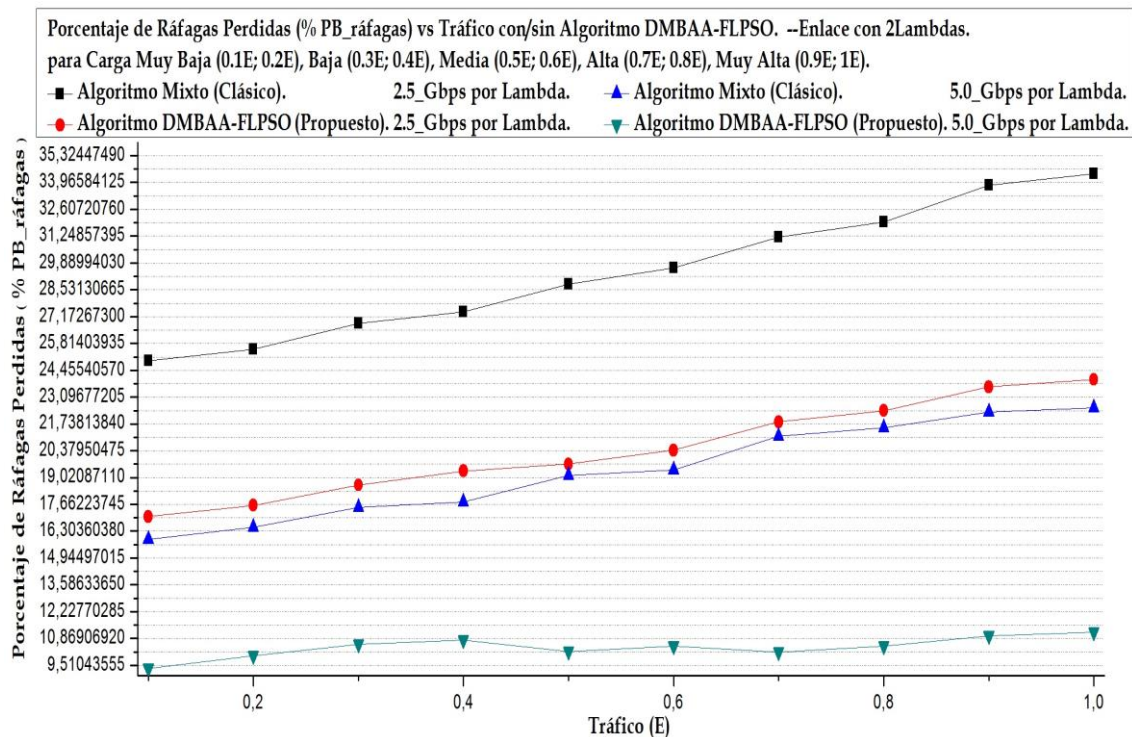


Figura B.3 Porcentaje de probabilidad de bloqueo (% PB) vs tráfico (variable), para 2.5 Gbps y 5.0 Gbps por λ .



B.4 Resultados obtenidos del porcentaje del retardo extremo a extremo (% e2e-D), en función del tráfico de entrada variable a una red OBS distribuida NSFNet, para 2.5 Gbps y 5.0 Gbps por Lambda, con 2 Lambdas (para carga útil –ráfagas) por Enlace de Fibra (DWDM).

Tabla B.4 Porcentaje del retardo extremo a extremo (% e2e-D) vs Tráfico (variable) para 2.5 Gbps y 5.0 Gbps por Lambda.

E	%Retardo extremo a extremo (%e2e-D) Para 2.5 Gbps por λ . Con 2 λ por enlace.		%Retardo extremo a extremo (%e2e-D) Para 5.0 Gbps por λ . Con 2 λ por enlace.	
	Algoritmo Mixto (Clásico)	Algoritmo DMBAА-FLPSO (Propuesto)	Algoritmo Mixto (Clásico)	Algoritmo DMBAА-FLPSO (Propuesto)
1	0,4053009030	0,4053990581	0,40513288010913	0,405166493524207
0,9	0,4052885444	0,4053980085	0,40513219802361	0,405166010686492
0,8	0,4052636407	0,4053777571	0,40513022348238	0,405162810795124
0,7	0,4052537717	0,4053230380	0,40512950206776	0,405159074122462
0,6	0,4052371828	0,4052835477	0,40512818320663	0,405146714973804
0,5	0,4052295608	0,4052632186	0,40512743309023	0,405145498038630
0,4	0,4052173372	0,4052179996	0,40512642379294	0,405137297808158
0,3	0,4052126250	0,4052131228	0,40512607880431	0,405136330540948
0,2	0,4052015042	0,4052027306	0,40512481153725	0,405134745645269
0,1	0,4051980319	0,4051994564	0,40512443515570	0,405133401330078

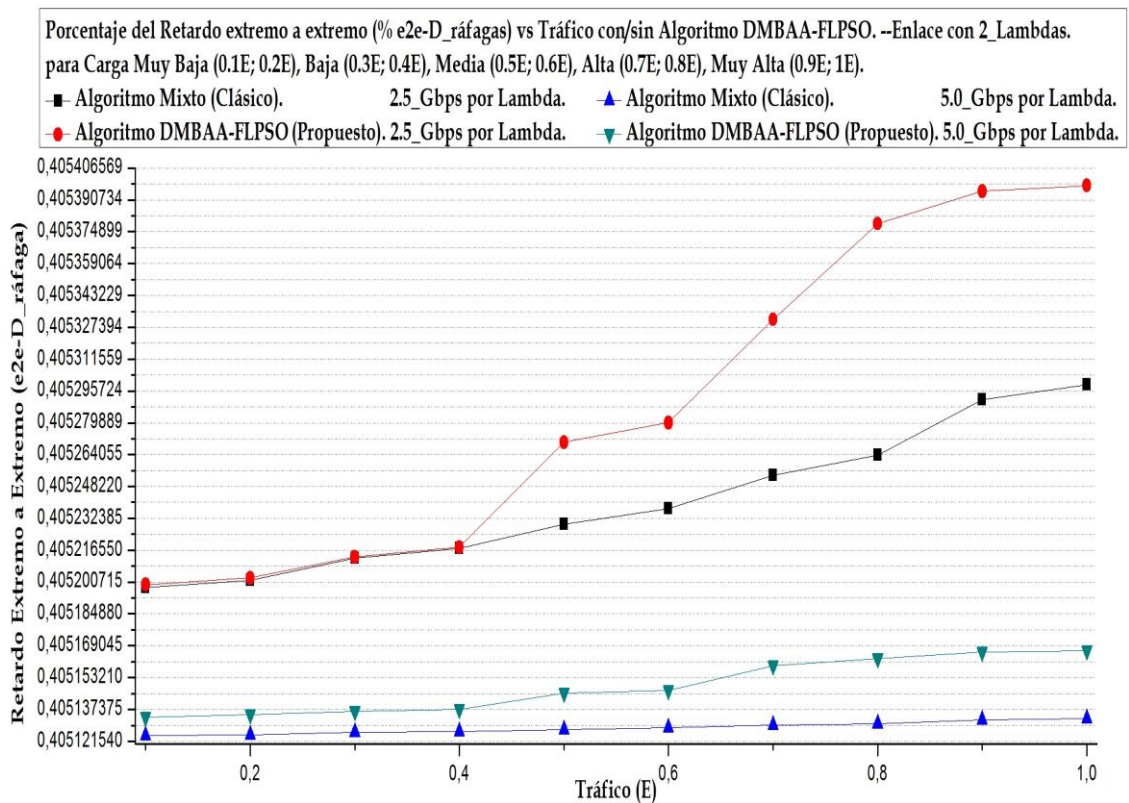


Figura B.4 Porcentaje de probabilidad de bloqueo (% PB) vs tráfico (variable), para 2.5 Gbps y 5.0 Gbps por λ .