

**UNIVERSIDAD NACIONAL DE CHIMBORAZO**



**FACULTAD DE INGENIERÍA**

**CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

**Proyecto de Investigación previo a la obtención del título de: Ingeniero  
en Electrónica y Telecomunicaciones**

**TRABAJO DE TITULACIÓN**

**“DISEÑO Y SIMULACIÓN DE INTSERV6 SOBRE EL PROTOCOLO  
HMIPV6-BI PARA OPTIMIZAR EL RENDIMIENTO EN LAS REDES  
MÓVILES IPV6”**

**Autora:**

Gabriela Katerine López Rodas

**Tutor:**

Mgs. Deysi Vilma Inca Balseca

**Riobamba - Ecuador**

**Año 2020**

Los miembros del tribunal de graduación del proyecto de investigación de título: **“DISEÑO Y SIMULACIÓN DE INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI PARA OPTIMIZAR EL RENDIMIENTO EN LAS REDES MÓVILES IPV6”**, presentado por **Gabriela Katerine López Rodas** y dirigida por la **Mgs. Deysi Vilma Inca Balseca**.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual consta el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la Biblioteca de la Facultad de Ingeniería de la UNACH.

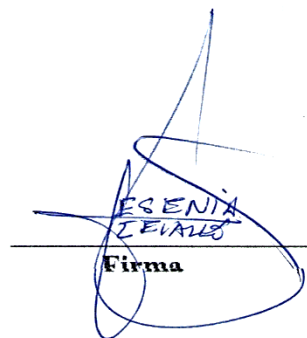
Para constancia de lo expuesto firman:

PhD. Leonardo Rentería  
**Presidente del Tribunal**



**Firma**

PhD. Yesenia Cevallos  
**Miembro del Tribunal**



**Firma**

MSc. Alejandra Pozo  
**Miembro de Tribunal**



**Firma**

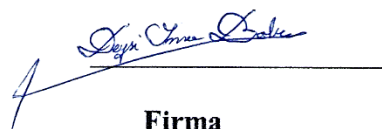
## **DECLARACIÓN EXPUESTA DE TUTORÍA**

En calidad de tutor del tema de investigación: **“DISEÑO Y SIMULACIÓN DE INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI PARA OPTIMIZAR EL RENDIMIENTO EN LAS REDES MÓVILES IPV6”**. Realizado por la Srta. Gabriela Katerine López Rodas, para optar por el título de Ingeniero en Electrónica y Telecomunicaciones, considero que reúne los requisitos y méritos suficientes para ser sustentada públicamente para ser sustentada públicamente y evaluada por el jurado examinador que se designe.

Riobamba, enero 2020.

Mgs. Deysi Inca Balseca  
C.I. 060381048-2

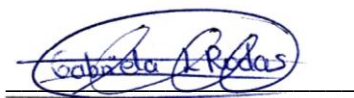
**Tutora**



**Firma**

## AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este proyecto de graduación titulado “**DISEÑO Y SIMULACIÓN DE INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI PARA OPTIMIZAR EL RENDIMIENTO EN LAS REDES MÓVILES IPV6**”, corresponde exclusivamente a Gabriela Katerine López Rodas y Mgs. Deysi Inca Balseca; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

A handwritten signature in blue ink, enclosed in a blue oval. The signature appears to read 'Gabriela K. Rodas'.

Gabriela Katerine López Rodas

C.I 0605497171

## **DEDICATORIA**

El presente trabajo de titulación está dedicado primeramente a Dios, ya que gracias a él puedo culminar con éxito esta etapa de mi vida, etapa con grandes aprendizajes.

Gracias a Dios por mis hijos Isaac y Matías, posiblemente en este momento no sean capaces de entender lo que significan para mí, pero son la razón por la que cada día me levante a esforzarme por el presente y mi principal motivación para luchar en búsqueda de un mejor futuro para ustedes. Aun a su corta edad me han enseñado a amar incondicionalmente y le han dado un verdadero significado a mi vida por eso este logro se los dedico a ustedes.

De igual manera se la dedico a toda mi familia en especial a mi madre Doraliza Rodas y mi hermano Jorge López, que han sido las personas que siempre han creído en mí incluso en momentos difíciles; así como también en memoria de mi abuelito Isaac Rodas que desde el cielo me cuida y me guía.

***Gabriela López Rodas***

## **AGRADECIMIENTO**

Gracias a mi madre por ser la principal promotora de mis sueños, gracias por cada día confiar en mis expectativas, por siempre estar dispuesta a acompañarme cada larga y agotadora noche de estudio, gracias a mi hermano por anhelar siempre lo mejor para mí, gracias por cada consejo, palabra y acción que me guiaron durante mi vida.

Gracias a toda mi familia por siempre anhelar lo mejor para mí y para mis hijos, gracias por su apoyo y amor incondicional.

Durante el proceso del desarrollo de mi trabajo de titulación quiero agradecer a mis hijos Isaac y Matías, por cada beso amoroso y abrazo de apoyo para continuar en este proceso en búsqueda de obtener mi título profesional.

De igual manera agradezco a mi tutora por su apoyo y guía en desarrollo de mi trabajo de titulación.

***Gabriela López Rodas***

## ÍNDICE GENERAL

ÍNDICE DE FIGURAS .....	X
ÍNDICE DE TABLAS .....	XI
RESUMEN .....	XII
ABSTRACT .....	XIII
INTRODUCCIÓN .....	1
CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA .....	3
1.1. PROBLEMA Y JUSTIFICACIÓN .....	3
1.2. OBJETIVOS .....	4
1.2.1. OBJETIVO GENERAL .....	4
1.2.2. OBJETIVOS ESPECÍFICOS .....	4
CAPÍTULO II. MARCO TEÓRICO .....	5
1.3. ANTECEDENTES .....	5
1.4. PROTOCOLO IPV6 .....	6
1.4.1. Direccionamiento IPv6 .....	7
1.4.2. Alcance de las direcciones IPv6 .....	7
1.5. MOVILIDAD DE REDES MÓVILES IPV6 .....	7
1.5.1. Concepto de Movilidad .....	7
1.5.2. Redes Móviles e IPv6 .....	8
1.6. PROTOCOLOS DE MOVILIDAD HMIPV6 Y HMIPV6-BI .....	11
1.6.1. Protocolo HMIPv6 .....	11
1.6.2. HMIPv6-BI .....	12
1.6.3. Estructura de datos .....	13
1.6.4. Operaciones de los elementos de red .....	13
1.7. INTEGRACIÓN DE SERVICIOS VERSIÓN 6 .....	15
1.8. NETWORK SIMULATOR VERSIÓN3 (NS3) .....	16

1.8.1.	Prerrequisitos para la instalación de NS3 .....	17
1.8.2.	Instalación de NS3.....	19
CAPÍTULO III. METODOLOGÍA DE LA INVESTIGACIÓN .....		21
3.1.	TIPO Y DISEÑO DE INVESTIGACIÓN.....	21
3.1.1.	Investigación Exploratoria.....	21
3.1.2.	Investigación Experimental .....	21
3.2.	MÉTODOS, TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN .....	21
3.2.1.	Método Científico.....	21
3.2.2.	Técnica de recolección de datos. ....	21
3.2.3.	Instrumentos .....	22
3.3.	POBLACIÓN Y MUESTRA.....	22
3.3.1.	POBLACIÓN .....	22
3.3.2.	MUESTRA .....	22
3.4.	HIPOTESIS.....	22
3.4.1.	OPERACIONALIZACIÓN DE LAS VARIABLES .....	23
3.5.	PROCEDIMIENTOS Y ANÁLISIS.....	23
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN .....		32
4.1.	SIMULACIONES.....	32
4.1.1.	SIMULACIÓN DE LA RED MÓVIL CON EL PROTOCOLO HMIPV6 32	
4.1.2.	SIMULACIÓN DE LA RED MÓVIL CON INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI.....	36
4.2.	ANÁLISIS .....	40
4.2.1.	Análisis de tiempos de comunicación. ....	40
4.2.2.	Análisis del BER. ....	40
4.2.3.	Análisis del Ancho de Banda.....	41
4.2.4.	Análisis de Latencia de Handover .....	42
4.3.	ANÁLISIS ESTADÍSTICO.....	42



4.3.1. Análisis estadístico de Throughput.....	44
4.3.2. Análisis estadístico de Jitter .....	45
4.3.3. Análisis estadístico de pérdida de paquetes.....	46
4.4. DISCUSIÓN .....	47
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....	48
5.1. CONCLUSIONES .....	48
5.2. RECOMENDACIONES.....	49
BIBLIOGRAFÍA .....	50
GLOSARIO.....	53
ANEXOS .....	54
ANEXO 1 – DESARROLLO DEL SCRIPT EN NS3 .....	54
ANEXO 2 - RED CON PROTOCOLO HMIPV6.....	59
ANEXO 3 - RED CON INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI.....	63
ANEXO 4 – SIMULACIÓN DE LA RED MÓVIL CON EL PROTOCOLO HMIPV6 .....	71
ANEXO 5 – SIMULACIÓN DE LA RED MÓVIL CON INTSERV6 SOBRE PROTOCOLO HMIPV6-BI.....	76

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Direccionamiento IPV6.....	7
<b>Figura 2.</b> Funcionalidad básica de una red móvil.....	8
<b>Figura 3.</b> Arquitectura GPRS y SAE.....	10
<b>Figura 4.</b> Pila de protocolos de la arquitectura GPRS. ....	10
<b>Figura 5.</b> Arquitectura de una red HMIPV6.....	12
<b>Figura 6.</b> Arquitectura HMIPV6 y HMIPV6-BI .....	13
<b>Figura 7.</b> . Descubrimiento del MAP.....	14
<b>Figura 8.</b> Mensajes de actualización y registro de MN .....	14
<b>Figura 9.</b> Hashing en IntServ.....	15
<b>Figura 10.</b> Clasificación de paquetes de un router InsServ e InsServ6 .....	16
<b>Figura 11.</b> Network Simulator 3.....	17
<b>Figura 12.</b> Instalación de NS3- Pasos.....	19
<b>Figura 13.</b> Bibliotecas de NS-3 .....	24
<b>Figura 14.</b> Estructura de un script.....	25
<b>Figura 15.</b> Diagrama de flujo para el desarrollo del proyecto .....	25
<b>Figura 16.</b> Topología para el escenario1 .....	26
<b>Figura 17.</b> Diagrama de flujo para implementar el escenario1. ....	27
<b>Figura 18.</b> Topología para implementar del protocolo HMIPv6-BI.....	28
<b>Figura 19.</b> Diagrama de Flujo para implementar el protocolo HMIPv6-BI.....	29
<b>Figura 20.</b> Como funciona IntServ6 .....	31
<b>Figura 21.</b> Comunicación entre CN y MN1 del escenario1 .....	32
<b>Figura 22.</b> Fórmula para calcular el BER.....	35
<b>Figura 23.</b> Comparación de tiempo de comunicación entre el escenario1 y escenario final .....	40
<b>Figura 24.</b> Comparación del BER entre el escenario1 y escenario final.....	41
<b>Figura 25.</b> Comparación del Ancho de Banda entre el escenario1 y escenario final ....	41
<b>Figura 26.</b> Comparación de latencia de handover entre el escenario1 y escenario final42	
<b>Figura 27.</b> Prueba de normalidad con IBM SPSS .....	43
<b>Figura 28.</b> Datos descriptivos de tiempos de conexión con IBM SPSS.....	44
<b>Figura 29.</b> Contraste hipótesis de Throughput con IBM SPSS .....	45
<b>Figura 30.</b> Contraste hipótesis de Throughput con IBM SPSS .....	46
<b>Figura 31.</b> Contraste hipótesis de Throughput con IBM SPSS .....	46

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Terminología de IPV6 .....	6
<b>Tabla 2.</b> Comparación entre redes 3G y 4G .....	8
<b>Tabla 3.</b> Componentes de la Arquitectura GPRS Y SAE .....	9
<b>Tabla 4.</b> Terminología de HMIPV6.....	11
<b>Tabla 5.</b> Prerrequisitos para la instalación de NS3 .....	18
<b>Tabla 6.</b> Comandos para la instalación de NS3 .....	19
<b>Tabla 7.</b> Descripción de módulos de inclusión para el escenario final.....	30
<b>Tabla 8.</b> Tiempos de conexión entre CN-MN1 .....	34
<b>Tabla 9.</b> Medición del Throughput del escenario1 .....	34
<b>Tabla 10.</b> BER del escenario1 .....	35
<b>Tabla 11.</b> JITTER del escenario1 .....	35
<b>Tabla 12.</b> Pérdida de paquetes del escenario1 .....	36
<b>Tabla 13.</b> Tiempos de conexión bidireccional entre CN-MN1.....	37
<b>Tabla 14.</b> Throughput del escenario final.....	38
<b>Tabla 15.</b> BER del escenario final .....	38
<b>Tabla 16.</b> Jitter del escenario final.....	39
<b>Tabla 17.</b> Pérdida de paquetes del escenario final.....	39
<b>Tabla 18.</b> Comparación de los tiempos de conexión entre escenario1 y escenario final40	

## RESUMEN

La utilización de equipos y dispositivos móviles en la actualidad es muy usual, ya que permiten tener acceso a la red inalámbrica de internet para mantener una comunicación y transferir información. Para mantener esta conexión a internet en micromovilidad existe el protocolo HMIPv6 (Hierarchical Mobile IPv6) mismo que realiza varios pasos para establecer la conexión entre el router de acceso (AR) y el nodo móvil (MN) con el que se quiere comunicar el nodo correspondiente y su modificación para aprovechar el ancho de banda es HMIPv6-BI (Improve the Radio Channel Bandwidth in Hierarchical Mobile IPv6).

El presente trabajo de investigación se diseñó una topología para la simulación del protocolo HMIPv6-BI que disminuye la utilización el ancho de banda en la comunicación inalámbrica entre AR y MN, posteriormente se brinda calidad de servicio mediante la implementación de IntServ6 cuya función principal radica en el cálculo del clave hash en el host origen cada que llegan paquetes de la red hacia el router que realiza la clasificación de paquetes en el simulador de redes NS-3.

NS-3 permite mediante las simulaciones recolectar datos de los parámetros necesarios para determinar el rendimiento de la red como son: Throughput, Jitter, BER, Perdida de paquetes, Ancho de Banda y Latencia, posteriormente se comparó y se realizó un análisis estadístico de dichos valores con los de una red que no utiliza estos protocolos. Comparaciones que permiten validar la funcionalidad de IntServ6 sobre el protocolo HMIPv6-BI.

**Palabras Clave:** calidad de servicio, HMIPv6, HMIPv6-BI, IntServ6, IPv6, movilidad, rendimiento.

## ABSTRACT

### ABSTRACT

Nowadays, it is common for people the usage of mobile devices since they allow people access to the wireless internet network to maintain communication and transfer information. There is the HMIPv6 (Hierarchical Mobile IPv6) protocol to keep the internet connection to micro mobility. It performs several steps to establish the connection between the access router (AR) and the mobile node (MN) with which the corresponding node is to communicate, and its modification to take advantage of bandwidth is HMIPv6-BI (Improve the Radio Channel Bandwidth in Hierarchical Mobile IPv6). This research presents a topology for the simulation of the HMIPv6-BI protocol to reduces the use of bandwidth in wireless communication between AR and MN. Consequently, quality of service is provided using the implementation of IntServ6 whose primary function lies in the calculation of the hash key on the source host each time packets from the network arrive towards the router that performs the classification of packets in the NS-3 network simulator. NS-3 allows simulations to collect data on the parameters necessary to determine the performance of the network, such as Throughput, Jitter, BER, packet loss, bandwidth, and latency. It was compared and statistical analysis of said values with those of a network that does not use these protocols. These comparisons allow validating the functionality of IntServ6 on the HMIPv6-BI contract.

**KEYWORDS:** Quality of service, HMIPv6, HMIPv6-BI, IntServ6, IPv6, mobility, performance.



Reviewed by: Guaranga, Jessica



LINGUISTIC COMPETENCES TEACHER

## INTRODUCCIÓN

En la actualidad los sistemas de telecomunicaciones está direccionando su uso e investigaciones al protocolo IPv6 (El Ksimi & Leghris, 2018) para mejorar el aspecto de movilidad (Díaz, Demasi, Robles, & Vodopivec, 2006) se crearon los protocolos MIPv6 tecnología que fue diseñada para solucionar problemas de macro movilidad. Mientras que HMIPv6 (Patil & Patil, 2010) es el protocolo más utilizado y sobre el cual se están desarrollando las investigaciones ya que proporciona varias beneficios como: disminución de latencia de handover, una mejor disposición y desempeño de la red, buscar un reenrutamiento eficiente y mejorar la calidad de servicio en video. Los usuarios utilizan el roaming cuando se desplazan viajando a través de diferentes redes, el principal problema es que al cambiarse de red se establece una dirección IP diferente a la de la red anterior con la que establecía una conexión, por lo que el usuario no puede tener una sesión de red abierta durante la movilización y el desplazamiento. La IETF desarrolló un nuevo modelo de conectividad a Internet que busca la solución a los problemas de conectividad y movilidad mencionados con anterioridad, pero que solo es aceptable sobre IPv6 (Castillo Medina & Forero Rodríguez, 2013)

IPv6 para lograr el enlace entre dos nodos, cada uno de los nodos realiza una determinada función, al recibir paquetes un nodo debe tunelizar, dicho proceso es conocido como encapsulamiento IPv6, mientras que el otro nodo que se encarga de desencapsular los paquetes que se enviaron a su destino.

Para que los paquetes lleguen a su destino final el protocolo HMIPv6-BI (Becerra, Padilla, & Paradells, 2011) tiene como propósito mejorar el uso del ancho de banda evitando el desperdicio por el aumento de bytes durante en enlace de comunicación entre los nodos IPv6 como lo hace el protocolo antecesor HMIPv6 y mediante el uso de IntServ6 se busca mejorar el tiempo de procesamiento de la información, velocidad de transmisión y la latencia de handover para mejorar el rendimiento de la red.

Por lo tanto, en el presente proyecto se presentará la investigación para mejorar el rendimiento de las redes IPv6, desarrollando una simulación de redes móviles de comunicaciones con el protocolo HMIPv6-BI y posteriormente aplicando IntServ6

(Padilla et al., 2005) sobre el protocolo anterior mencionado, como primer punto una simulación previa que consista en redes móviles IPv6 con HMIPv6, luego con el uso del protocolo HMIPv6-BI, comparar con los indicadores de la infraestructura de red y como último paso aplicar IntServ6 sobre el protocolo HMIPv6-BI que se consideraría el modelo de simulación final y sobre el cual se va a realizar las mediciones en un software que permita medir los parámetros de redes de comunicación para verificar su rendimiento.

## **CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA**

### **1.1. PROBLEMA Y JUSTIFICACIÓN**

Para poder tener acceso a la red internet y mantener una conexión para el traspaso de información es indispensable tener una red que soporte protocolos de movilidad para brindar mejor servicio y al mismo optimizar el rendimiento de acuerdo a los parámetros como lo son: Throughput, Jitter, BER (BIT ERROR RATE), pérdida de paquetes para validar la funcionalidad de la red móvil.

Los problemas detectados en las redes móviles son:

- En el uso del ancho de banda que se desperdicia, al realizar varios pasos para establecer la conexión entre los nodos de red.
- En el momento cuando se detecta una mayor carga de tráfico, no se puede establecer el tiempo correcto para que el protocolo MAP detecte y envíe la nueva ubicación de MN y por lo tanto no se pueda determinar el tipo de handover (suave y duro).
- Otro problema en IPV6 es la falta de garantías en cuanto a calidad de servicio que se ofrece solo mediante. Además, los tiempos de retardo son considerables para la transmisión de paquetes.

Por lo anteriormente mencionada al implementar en la red IntServ6 se busca mejorar la calidad de servicio mediante la reducción de tiempos y procesos para transferencia de paquetes y también al implementarlo sobre el protocolo HMIPv6-BI se pretende mejorar el ancho de banda de la red móvil.



## **1.2. OBJETIVOS**

### **1.2.1. OBJETIVO GENERAL**

Diseñar y simular IntServ6 sobre el protocolo HMIPv6-BI mediante una herramienta de software para mejorar la velocidad de transmisión de paquetes, latencia de handover y la optimización del ancho de banda de redes móviles IPv6

### **1.2.2. OBJETIVOS ESPECÍFICOS**

- Diseñar y simular un escenario preliminar, mediante una herramienta de software para determinar el rendimiento y problemas de la red móvil IPv6.
- Diseñar y simular redes móviles implementando el protocolo HMIPv6-BI para validar su funcionamiento en la optimización del ancho de banda.
- Implementar IntServ6 sobre el protocolo HMIPv6-BI que permita realizar cambios o modificaciones en un escenario de trabajo.
- Comparar y analizar el rendimiento del escenario inicial y el escenario de trabajo final mediante indicadores de infraestructura de red como son: Throughput, Jitter, BER (BIT ERROR RATE), pérdida de paquetes para validar el funcionamiento del protocolo IntServ6 sobre HMIPv6-BI.

## **CAPÍTULO II. MARCO TEÓRICO**

### **2.1. ANTECEDENTES**

La utilización de equipos y dispositivos móviles en la actualidad es muy usual, ya que permiten tener acceso a la red de internet y así lograr mantener una comunicación con otras personas u organizaciones para recibir o transferir información durante la comunicación para mantener esta conexión a internet existe el protocolo MIPv6. La funcionalidad del protocolo MIPv6 se basa en mantener una conexión permanente a Internet incluso cuando el dispositivo este fuera del área local resolviendo los problemas de macro movilidad y para problemas generados respecto a la micromovilidad se creó el protocolo HMIPv6.

HMIPv6(Zhang & Bi, 2012) agrega un nivel al protocolo MIPv6, para poder diferenciar que tipo de movilidad que se tiene ya sea esta local o global. Si son problemas de micromovilidad, la solución se incrementa un nodo nuevo llamando Mobility Anchor Point (MAP), este nuevo nodo es que gestiona la movilidad cada vez que el Nodo Correspondiente (CN) envía un paquete al Nodo Móvil (MN), MAP lo intercepta y tuneliza hacia el Nodo Móvil. El MN es quien quita la cabecera del túnel, procesa el paquete y es aquel que se mueve dentro del dominio de MAP. Cuando el MAP cambie de dominio igual que su Home Agent (HA); el Nodo Móvil y Nodo Correspondiente deben actualizarse también; todo este proceso permite la reducción de la señalización entre MN, CN y HA permitiendo de esta manera la comunicación sin interrupciones ni pérdida de paquetes, dando mayor eficiencia al protocolo HMIPv6.

HMIPv6-BI es una modificación del protocolo que gestiona la micromovilidad HMIPv6 con el objetivo de reducir y optimizar el ancho de banda como también la latencia de handover para lograr estos beneficios el túnel llega sólo hasta el Router de Acceso (AR), no como lo hace HMIPv6 que llega hasta el MN que hace un túnel con MAP provocando números adiciones de bytes para que viajen por el aire (Becerra, Padilla, & Paradells, 2011).

Las redes móviles también necesitan proveer calidad de servicio (Evaluación de IntServ6: una propuesta de calidad del servicio en redes IPv6, 2017), para soportar Calidad del Servicio extremo a extremo en una red Internet que trabaja con IPv6 se utiliza las etiquetas de flujo de IPv6 para mejorar el desempeño en el proceso de clasificación de paquetes de los Routers IntServ, para una arquitectura del tipo IntServ6 el proceso de clasificación se reduce a 2 pasos:

1. **Lectura de la cabecera** la lectura de la cabecera se hace con el fin de obtener el número HASH calculado en el origen
2. **Búsqueda de la reserva.** y para llevar a cabo la búsqueda de la reserva de flujo en la tabla HASH es necesaria una sola iteración debido a que la posición de la reserva es apuntada por el número HASH.

Mediante el software de simulación de redes NS-3 (Hugo, Espinoza, Enrique, & Lovato, n.d.) se realiza scripts de las topologías de red que utilizan protocolos de micromovilidad como también la propuesta de IntServ6 para brindar calidad de servicio a la red y de esta manera validar su funcionamiento.

## 2.2.PROTOCOLO IPV6

El protocolo IPV6 es el sucesor del protocolo IPv4, una de las principales ventajas es las direcciones libres que posee y que no es necesario recurrir a recursos como NAT (Network Address Translation) para mantener la conectividad, debido que el tamaño de direcciones IP aumento de 32bits de IPV4 a 128 bits. Además, el diseño del protocolo soporta seguridad, calidad de servicio, facilidad de administración de acuerdo al RFC 2460 se generó la **Tabla 1** y se explica a mayor detalle la terminología que utiliza el protocolo IPv6 (Mejia, 2011) .

*Tabla 1. Terminología de IPV6*

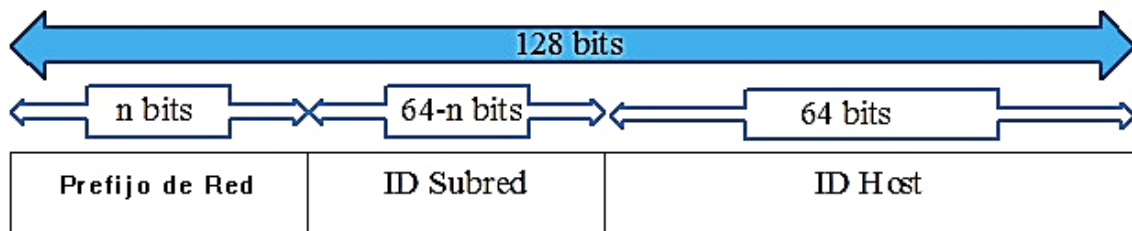
<b>TÉRMINO</b>	<b>DEFINICIÓN</b>
NODO	Dispositivo implementado en IPV6
ENRUTADOR	Nodo que reenvía paquetes
HOST	Nodo que no se comporta como enrutador
CAPA SUPERIOR	Capa inmediata superior a IPV6
ENLACE	Una facilidad para la comunicación de nodo en la capa enlace
VECINOS	Nodos conectados al mismo enlace
INTERFACE	Permite acoplar un nodo a un enlace
DIRECCIÓN	Identificador en la capa IPV6
PAQUETE	Cabecera IPV6 más carga útil
MTU de enlace	Unidad de transmisión máxima
MTU de ruta	MTU de enlace mínima dentro de una ruta origen-destino

**Fuente:** RFC 2460

### 2.2.1. Direccionamiento IPv6

Una dirección IPv6 (Lee, 2011) está descrita por 128 bits distribuidos de la siguiente manera como se puede diferenciar en la **Figura 1**. Y se detallan a continuación:

- **Prefijo de red:** Sirve para identificar una red en específico.
- **Identificador de subred:** Realiza la función de identificar un enlace de un sitio.
- **Identificador de host:** Es el identificador único del nodo en la red a la que pertenece.



*Figura 1. Direccionamiento IPV6*

**Fuente:** (Lee, 2011)

### 2.2.2. Alcance de las direcciones IPv6

Cada interfaz tiene un alcance máximo debido a la ubicación geográfica y el alcance que tienen las direcciones IPv6 son las siguientes:

- **Link Local:** Redes de dirección local, igual al alcance de las red LAN (Network Area Local)
- **Site Local:** Estas direcciones permiten un alcance de un campus o una ciudad
- **Global:** Es la dirección que tiene cada interfaz en el internet

## 2.3.MOVILIDAD DE REDES MÓVILES IPV6

### 2.3.1. Concepto de Movilidad

La movilidad (Díaz et al., 2006)se refiere a la capacidad de mantener la conexión a internet sin interrupción e independientemente de la ubicación en que este o el cambio de punto de acceso que tenga a la red. Sin hacer autoconfiguraciones de la red sino aceptar las conexiones que vengan del exterior, es decir que los equipos parecen que no estuvieran en movimiento cuando incluso pueden estar en constante movimiento o cambiando entre varias redes (Fernández, Domínguez, Ipv, & Ipv, 2015).

### 2.3.2. Redes Móviles e IPv6

La necesidad de mantener la conexión entre dispositivos móviles o equipos M2M que usan la red para su comunicación, se realiza mediante la implementación de IPv6 que hace posible este crecimiento, ya que sin el protocolo antes mencionado se corre el riesgo de poder dar el servicio. Las redes móviles 3G y 4G están diseñadas para soportar el protocolo IPv6, la tecnología 3G es útil el esquema Dual Stack mientras que para la tecnología 4G es recomendable IPv6 solamente como se observa en la **Figura 2** se detalla la funcionalidad de una red móvil (Castillo Medina & Forero Rodríguez, 2013).



*Figura 2. Funcionalidad básica de una red móvil*

**Fuente:** (Castillo Medina & Forero Rodríguez, 2013)

En la **Tabla 2** se detallan las principales diferencias de redes 3G y 4G:

*Tabla 2. Comparación entre redes 3G y 4G*

GENERACIÓN	TECNOLOGÍA	CARÁCTERÍSTICAS	VELOCIDAD DE DESCARGA
3G	CDMA Y HSPDA	Streaming de música y video Mayor velocidad de navegación	2.4Mbps hasta 14.4 Mbps
4G	OFDMA, LTE, LTE Advanced	Streaming de video HD Videoconferencias en HD Llamadas de VoIP de alta calidad	100Mbps hasta 1Gbps

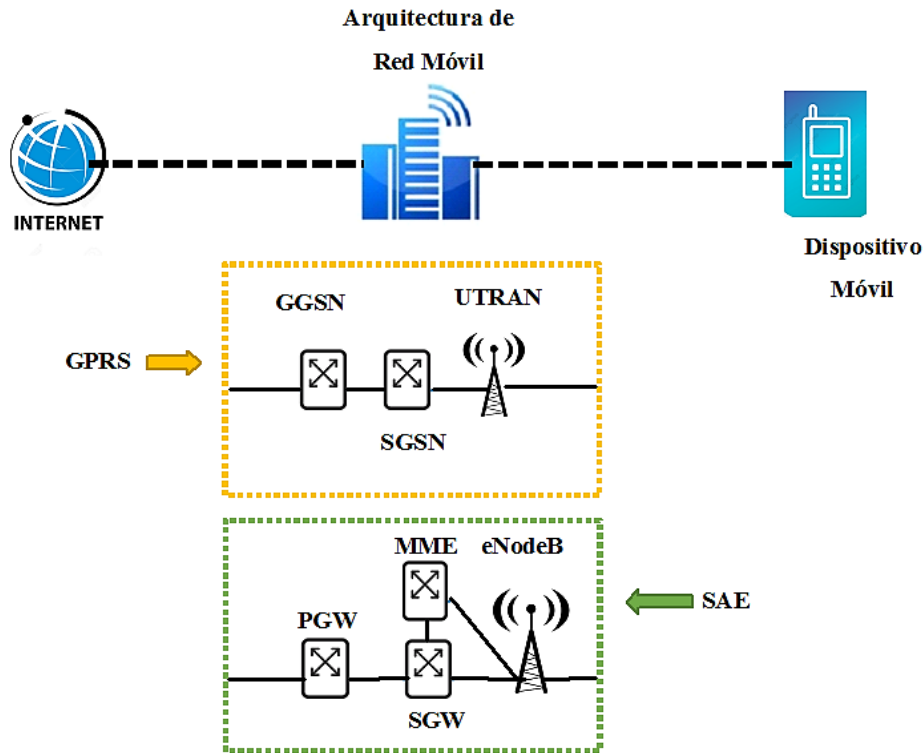
**Fuente:** (Despliegue de IPv6 en Redes Móviles Tomás Lynch Entusiasta de IPv6)

También las tecnologías de 3G y 4G se diferencian en la arquitectura que utilizan para la comunicación, como indica la **Figura 3**, la estructura de 3G es GPRS (General Packet Radio Service) y la estructura 4G es SAE (System arquitectura Evolution), que es la evolución de GPRS y se simplifica ya que solo usa IP. La 2 arquitectura tiene diferentes componentes (Castillo Medina & Forero Rodríguez, 2013) y cada componente tiene una función específica como se detalla en la **Tabla 3**.

**Tabla 3.** Componentes de la Arquitectura GPRS Y SAE

Arquitectura GPRS		Arquitectura SAE	
Componente	Descripción	Componente	Descripción
UTRAN	Constituido por Nodo B y RNC (red de radio) Red que transporta datos y voz	ENodeB	Evolución de Nodo B incorpora las capacidades de RNC
SGSN	Manejo de la movilidad y el ruteo de paquetes entre usuarios y GGSN	MME	Realiza el control de acceso a la red.
		SCW	Nodo que realiza el ruteo de paquetes y la movilidad.
GGSN	Actúa como la puerta de enlace entre la red móvil y otras redes.	PCW	Actúa como el GGSN y capacidades de filtrado, políticas de acceso como también análisis de paquetes

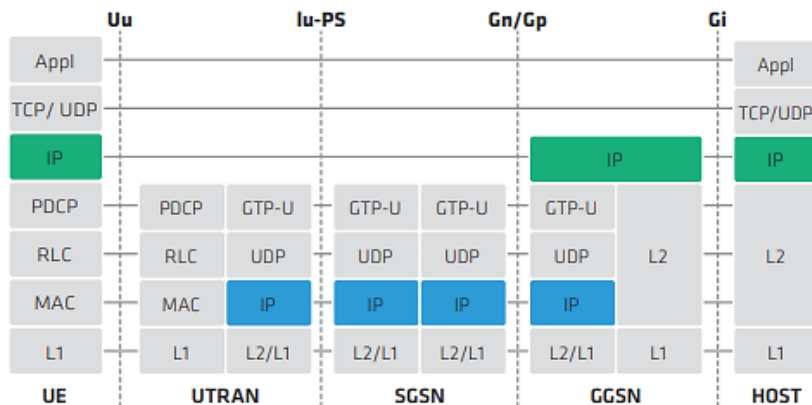
**Fuente:** (Castillo Medina, Felipe, Rodríguez, & Medina)



**Figura 3.** Arquitectura GPRS y SAE

**Fuente:** Autora, (Lynch & Moreiras, n.d.)

Tanto la arquitectura GPRS como SAE trabajan con el modelo OSI, para el proyecto de investigación se centrará en la capa 2, capa de enlace como se observa en la **Figura 4.** que es sobre la cual trabaja el protocolo IP para gestionar la movilidad de la red móvil, también se debe mencionar que la pila de protocolos de SAE solo varia en la funcionalidad que realizan los protocolos.



**Figura 4.** Pila de protocolos de la arquitectura GPRS.

**Fuente:** (Lynch & Moreiras, n.d.)

## 2.4.PROTOCOLOS DE MOVILIDAD HMIPV6 Y HMIPV6-BI

### 2.4.1. Protocolo HMIPv6

Para mejorar los problemas de micro movilidad en IPv6 se creó el protocolo HMIPv6 (Zhang & Bi, 2012) que permite a creación de una red jerárquica. Lo que realiza es el aumento de un nodo llamado MAP, para ayudar a mantener la conexión cuando un CN busca a un MN La terminología se detalla en la **Tabla 4**.

El proceso para lograr dicha conexión es la siguiente:

Si el CN envía un paquete hacia el MN

- MAP intercepta y tuneliza hacia una nueva ubicación nLCoA del MN y

Si el MN envía un paquete hacia el CN

- MAP retira la cabecera del túnel y reenvía la información original al CN

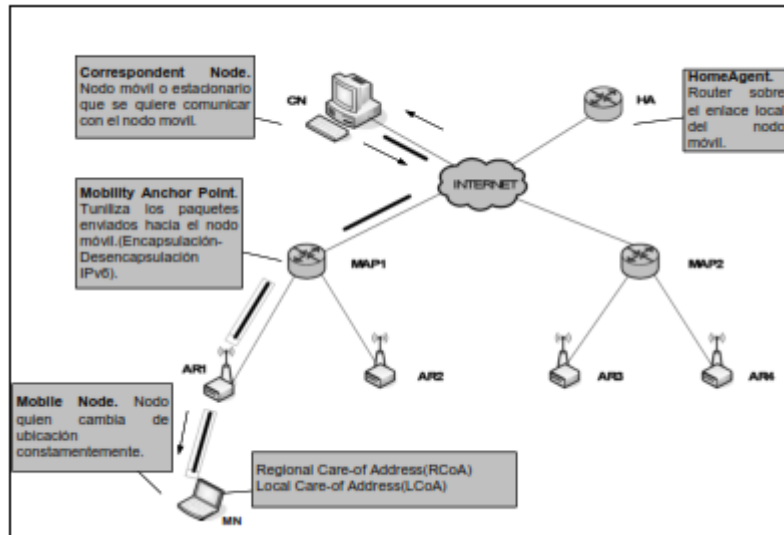
Como se muestra en la **Figura 5**, cuando el CN busca conectarse con el MN, la información va desde el CN hacia el Internet y el Punto de Anclaje móvil; y se realiza una tunelización bidireccional para la comunicación entre MAP-AR y AR-MN.

*Tabla 4. Terminología de HMIPv6*

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUNCIÓN</b>
<b>CN</b>	Nodo Correspondiente	Nodo que busca conexión con el MN.
<b>HA</b>	Agente Local	router sobre el enlace del nodo móvil.
<b>MAP</b>	Punto de Anclaje	enrutador ubicado en un red que ocupa un nodo móvil.
<b>AR</b>	Router de Acceso	Enrutador predeterminado del nodo móvil, agrega trafico saliente de nodos móviles.
<b>MN</b>	Nodo Móvil	Nodo que puede cambiar constantemente de ubicación.
<b>LCoA</b>	On-Link Care of Address (	Dirección del nodo móvil de una red.
<b>RCoA</b>	Regional Care of Address	Dirección del nodo móvil de una red visitada.
<b>LBU</b>	Local Binding Update	Actualización del enlace local al MAP para tener enlace entre RCoA y LCoA .

Fuente: RFC 4140





*Figura 5. Arquitectura de una red HMIPv6*

**Fuente:** (Becerra, Padilla, & Paradells, 2011)

#### 2.4.2. Protocolo HMIPv6-BI

HMIPv6-BI (Becerra, Padilla, & Paradells, 2011) es una modificación del protocolo HMIPv6, que lo que busca es eliminar el túnel que va desde en MN hacia el CN, para que el proceso de conexión sea el siguiente:

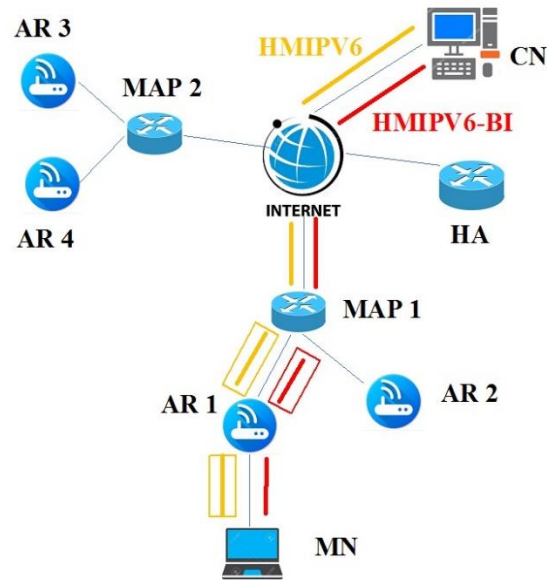
Si el CN envía un paquete hacia el MN:

- MAP intercepta y tuneliza hacia una AR.
- AR envía la información original al MN y

Si el MN envía el paquete original hacia el CN:

- MAP reenvía la información original al CN.
- 

Esta modificación lo que busca es reducir la carga de tráfico en el canal y así aumentar la eficiencia del canal, pero trabaja de igual forma que el protocolo HMIPv6. la **Figura 6**, muestra la comunicación del CN con el MN, la información va desde el CN hacia el Internet y el Punto de Anclaje móvil. Diferenciando la comunicación entre MAP-AR y AR-MN que para HMIPv6 es túnel entre el MAP al MN, mientras que para HMIPv6-bi es un túnel recortado que va solo desde el MAP al AR.



**Figura 6.** Arquitectura HMIPv6 y HMIPv6-BI

**Fuente:** Autora

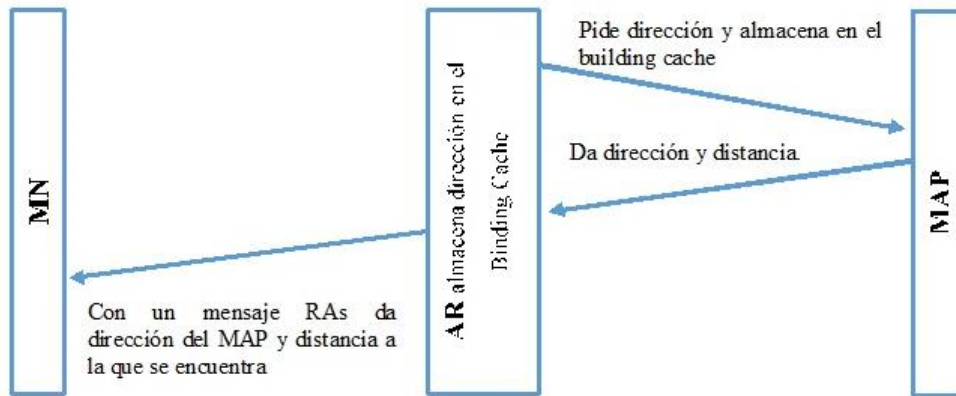
### 2.4.3. Estructura de datos

La estructura de datos (Díaz, Demasi, Robles, & Vodopivec) se clasifica en 3:

- **Binding Cache:** Es una tabla constituido por los enlaces actuales de los nodos móviles.
- **Binding Update List:** Registra actualizaciones de los nodos móviles.
- **Home Agents List:** Guarda información entre los agentes internos y el HA, comparte información con MAP.

### 2.4.4. Operaciones de los elementos de red.

**Descubrimiento del MAP:** Cuando el nodo móvil se encuentra en la red, mediante el AR se busca MAP que se encuentren y mediante un RAs (Router Advertisement) se envía al MN los datos que también se guarda en el Binding Cache. Ver **Figura 7**.

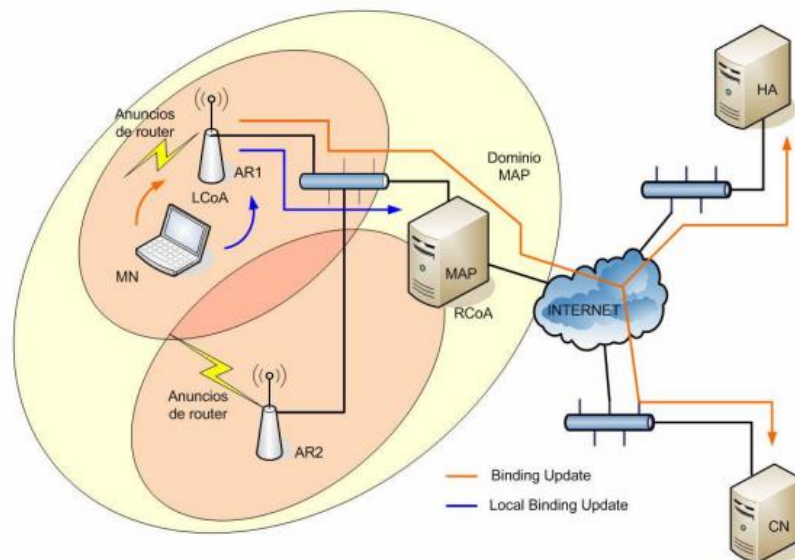


**Figura 7. . Descubrimiento del MAP**

**Fuente:** Autora

**Movimiento del MN:** Cuando el MN se mueve a la próxima red (Fernández et al., 2015), el MN solicita información del MAP para comparar si es que se encuentra en el mismo dominio de MAP o ya cambio de dominio y se procede de la siguiente manera:

- $MAP_{direcciónactual} = MAP_{direcciónanterior}$  ; entonces se envía un LBU.
- $MAP_{direcciónactual} \neq MAP_{direcciónanterior}$  ; entonces se envía un BU hacia el CN y HA.



**Figura 8. Mensajes de actualización y registro de MN**

**Fuente:** (Fernández et al., 2015)

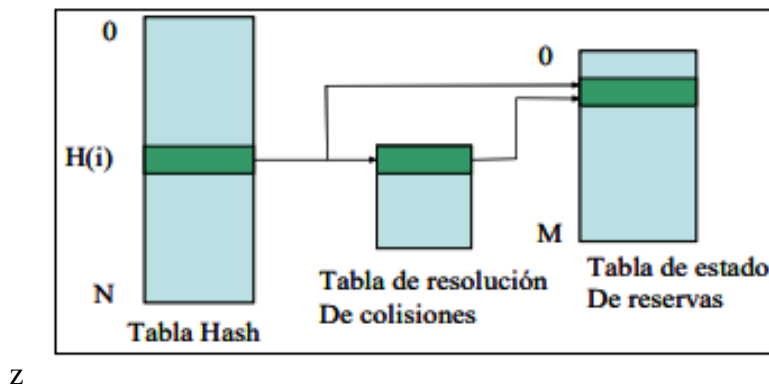
Para crear el túnel bidireccional entre MN y MAP se debe realizar el siguiente proceso:

- El MN envía el LBU al MAP
- MAP responde con Binding acknowledgement, sí o no.
- EL NM envía la nueva RCoA al HA, ver **Figura 8**.

## 2.5.INTEGRACIÓN DE SERVICIOS VERSIÓN 6

En las redes de comunicación se busca calidad de servicio para mejorar la red en general, para ello se plantea IntServ6 (Padilla, Paradells, & Rodríguez) como alternativa a desarrollar para que los router que soportan IntServ puedan realizar las tareas de la versión 6. El cálculo de la clave HASH en IntServ (Mahadevan & Sivalingam, 1999) se obtiene a partir de las direcciones de origen y destino, como también de los puertos tanto de origen como destino y la identificación del protocolo; todos estos elementos se denominan quintupla.

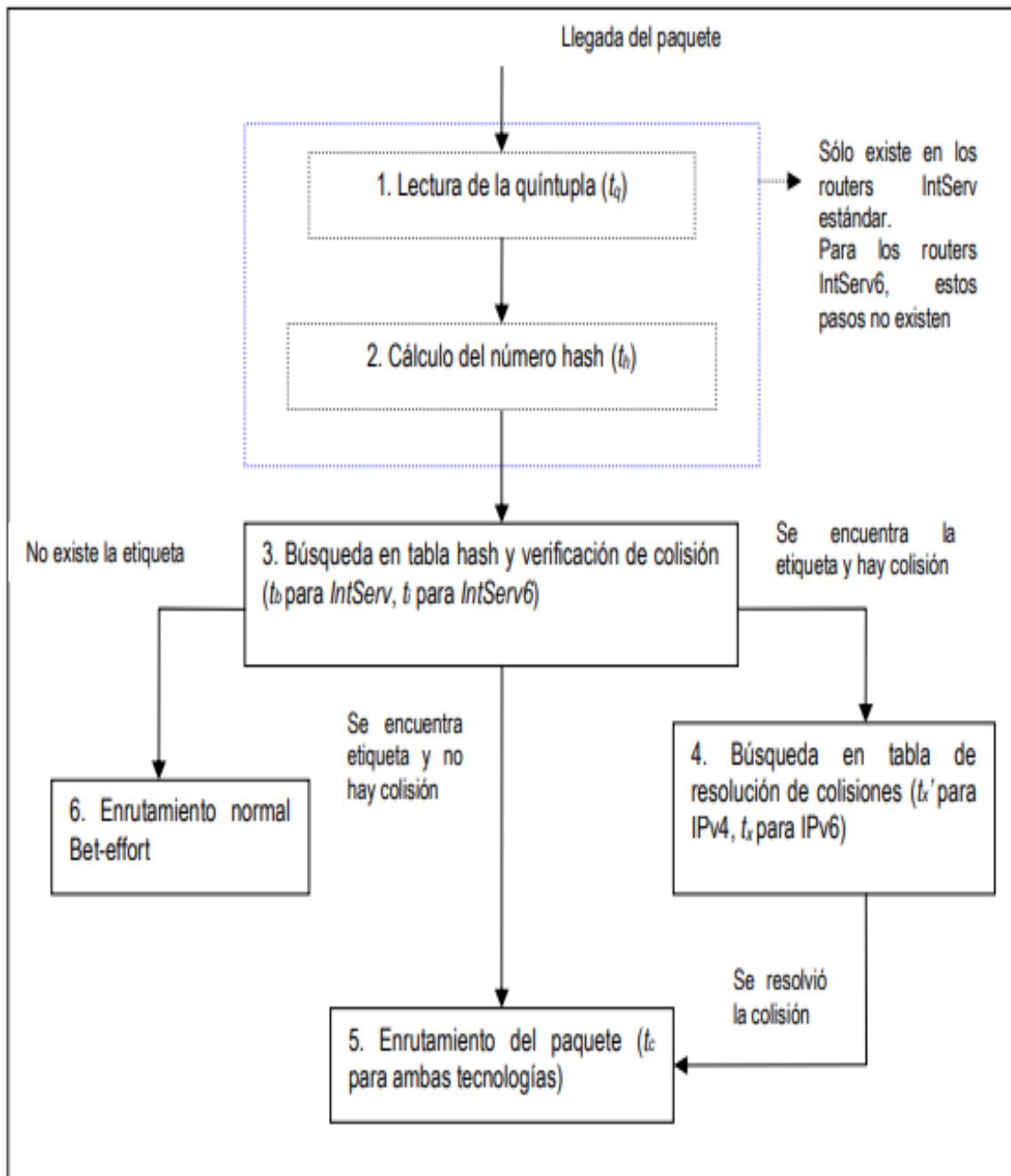
Este cálculo compara con la tabla HASH para verificar si el valor existe en la tabla de reservas de flujo y luego verificar que no tenga colisiones para que el número HASH apunte directo a la tabla de estado de reservas como muestra en la **Figura 9**. Si el valor tiene colisión se soluciona apuntando a la tabla de resoluciones de colisiones (Martinez Alayón, López, Ramírez Ochoa, & Gómez Tovar, 2013).



*Figura 9. Hashing en IntServ*

**Fuente:** (Martinez Alayón, López, Ramírez Ochoa, & Gómez Tovar, 2013)

La propuesta planteada de IntServ6 usa las etiquetas de flujo de IPV6 para agilizar el proceso de clasificación en los router, donde el cálculo de la clave HASH se realiza en el inicio. Se usa la clave HASH en la etiqueta de flujo para poder ser identificado, en la **Figura 10** se observa dicho proceso.



**Figura 10.** Clasificación de paquetes de un router *IntServ* e *IntServ6*

**Fuente:** (Padilla, Paradells, & Rodríguez)

## 2.6.NETWORK SIMULATOR VERSIÓN3 (NS3)

NS3 (ns-3 Tutorial Release ns-3.29 ns-3 project, 2019), es el software de simulación de código abierto para investigaciones en el desarrollo de nuevos protocolos, para validar o modificar protocolos ya existentes. También sirve para diseño y simulación de diferentes

topologías de redes de todo tipo sean estas: satelitales, inalámbricas, cableadas, entre otros. La infraestructura de NS-3 se basa en código C o C++, y permiten la visualización mediante el archivo de salida NAM (Network Animation) para poder analizar y procesar los datos obtenidos. Además, NS puede emular en tiempo real los procesos de redes, como también la verificación o modificación para la utilización de nuevos protocolos que optimicen las redes móviles.

Para poder instalar NS3 se requiere de la creación de una máquina virtual con el sistema operativo Ubuntu 18.04 sobre algún virtualizador, así como realizar una lista de prerequisites, que se muestra en la **Figura 11**.



*Figura 11. Network Simulator 3*

**Fuente:** Autora

### 2.6.1. Prerrequisitos para la instalación de NS3

Luego de la creación de la máquina virtual con UBUNTU 18.04 se debe realizar la instalación de paquetes necesarios como prerequisites que necesita NS3 (ns-3 Tutorial Release ns-3.29 ns-3 project, 2019) se detallan en la **Tabla 5**:

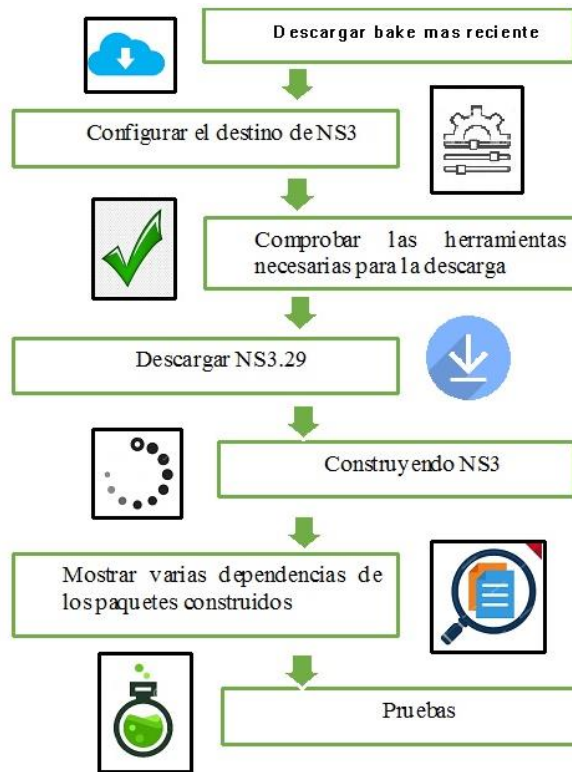
*Tabla 5. Prerrequisitos para la instalación de NS3*

PRERREQUISITOS	COMANDOS
Para C++	<code>sudo apt-get install gcc g++ python</code>
Para Python	<code>sudo apt-get install gcc g++ python python-dev</code>
Para desarrollo de Python	<code>sudo apt-get install Python-setuptools git mercurial</code>
Las herramientas de desarrollo qt4 son necesarias para Netanim Animator	<code>sudo apt-get install qt5-default mercurial</code>
para visualizador pyviz	<code>sudo apt-get install gir1.2-gooocanvas-2.0 python-gi python-gi-cairo python-pygraphviz python3-gi-cairo python3-pygraphviz gir1.2-gtk.3.0 ipython ipython3</code>
para emulación distribuida basada en MPI	<code>sudo apt-get install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev</code>
soporte del bake-herremientas	<code>sudo apt-get install autoconf cvs bzip unrar</code>
Herramientas de depuración, valgrind para depuración de memoria	<code>sudo apt-get install gdb valgrind</code>
para modificar el código fuente	<code>sudo apt-get install uncrustify</code>
para editar imagen y texlive para documentación	<code>sudo apt-get install doxygen graphviz imagemagick</code> <code>sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils texlive-lang-portuguese dvipng latexmk</code> <code>sudo apt-get install python-sphinx dia</code>
Soporte de la Biblioteca Científica GNU (GSL) para modelos de error WiFi más precisos	<code>sudo apt-get install gsl-bin libgsl-dev</code>
para analizador léxico flexible y generador de analizador de bisontes	<code>sudo apt-get install flex bison libfl-dev</code>
Para leer trazas de paquetes pcap	<code>sudo apt-get install tcpdump</code>
Soporte de bases de datos para el marco estadístico	<code>sudo apt-get install sqlite sqlite3 libsqlite3-dev</code>
para soporte de biblioteca Xml	<code>sudo apt-get install libxml2 libxml2-dev</code>
Paquetes de compilación	<code>sudo apt-get install cmake libc6-dev libc6-dev-i386 libclang-dev libnm-dev automake</code> <code>sudo apt install Python-pip</code> <code>sudo pip install cxxfilt</code>
para crear una interfaz gráfica de usuario	<code>sudo apt-get install libgtk2.0-0 libgtk2.0-dev</code>
para máquinas virtuales y ns-3	<code>sudo apt-get install vtun lxc</code>
para módulo de flujo abierto	<code>sudo apt-get install libboost-signals-dev libboost-filesystem-dev</code>

Fuente: Autora, [www.nsnam.org](http://www.nsnam.org)

### 2.6.2. Instalación de NS3

NS3 requiere algunos pasos para la correcta instalación como muestra la **Figura 12** y mediante la utilización de comandos que se verifican en la **Tabla 6**.



**Figura 12.** Instalación de NS3- Pasos

**Fuente:** Autora

Para la instalar NS-3 desde el terminal se detallan en la **Tabla 6** los comandos utilizados.

**Tabla 6.** Comandos para la instalación de NS3

<b>Descargar la copia más reciente del bake</b>	<b>cd</b>
	mkdir workspace
	cd workspace
	git clone https://gitlab.com/nsman/bake.git
<b>Configurar la versión destino</b>	cd bake
	./bake.py configure -e ns-allinone-3.29
<b>Confirmar las herramientas suficientes para descargar</b>	./bake.py check
<b>Descargar NS3</b>	./bake.py download
<b>Constuyendo NS3</b>	./bake.py build
	./bake.py show
<b>Prueba en NS3</b>	cd source/ns-3.29
	./test.py
	.waf --run hello-NS3

**Fuente:** Autora



NS3, ofrece también la posibilidad de instalar NetAnim para visualizar las redes y se instala de la siguiente manera:

1. hg clone <http://code.nsnam.org/netanim>
2. cd netanim
3. make clean
4. make

## **CAPÍTULO III. METODOLOGÍA DE LA INVESTIGACIÓN**

### **3.1. TIPO Y DISEÑO DE INVESTIGACIÓN**

#### **3.1.1. Investigación Exploratoria**

Este tipo de investigación es la primera que se va a llevar a cabo, para poder obtener toda la información relacionada con el tema de investigación mediante la búsqueda en documentos pasados. El objetivo de esta investigación es permitir tener una visión amplia acerca del tema y luego mediante un análisis de toda la información obtenida se generará nuevas preguntas para estudios posteriores (Edición Buenos Aires, 2006).

#### **3.1.2. Investigación Experimental**

La investigación experimental se basa en la recopilación de datos específicos que puedan ser de ayuda para dar las soluciones necesarias al tema de investigación. Y mediante el uso de variables experimentales aun no comprobadas que puedan ser manipuladas con la finalidad de llegar a una conclusión o decisión, después de realizar un respectivo análisis de las mismas (Edición Buenos Aires, 2006)

### **3.2.MÉTODOS, TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN**

#### **3.2.1. Método Científico**

El método científico es un procedimiento que tiene como objetivo establecer la relación que tienen diferentes hechos, para obtener un criterio fundamentado y comprobable sobre el tema de investigación. En este caso se podrá establecer criterios de infraestructura de la red para mejorar el rendimiento de la red IPv6 como son: Throughput, Jitter, BER, pérdida de paquetes. Así como calidad de servicio en IPV6 mediante la propuesta IntServ6

#### **3.2.2. Técnica de recolección de datos.**

La técnica de observación se basa en un conjunto de normas orientadas a evaluar un fenómeno en particular para conocer la realidad de dicho fenómeno, para detallar con precisión y de forma sistemática cualquier situación que se produzca de acuerdo a los objetivos planteados en la investigación. En este caso los parámetros seria: topología de red, tamaño de paquete, numero de paquete, rendimiento de red.

### **3.2.3. Instrumentos**

Los instrumentos para el desarrollo del presente documento son los siguientes:

- RFC 2460 Especificaciones del protocolo IPv6.
- RFC 3775 detalla el soporte de movilidad en IPv6.
- RFC 4140 que detallan el protocolo HMIPv6.
- Artículos relacionados a HMIPv6-BI e INTSERV6
- Manual de instalación y uso de NS3

## **3.3. POBLACIÓN Y MUESTRA**

### **3.3.1. POBLACIÓN**

La población fue infinita ya que las simulaciones de la red móvil con IntServ6 sobre el protocolo HMIPv6-BI no están limitadas en número ni en tiempo y se pueden hacer las veces que se requiera.

### **3.3.2. MUESTRA**

Para la evaluación de la red la muestra fue no probabilística intencional ya que para validar el rendimiento se analiza 3 parámetros como lo son: Jitter del cual se tomó 27 datos, Perdida de paquetes que se escogió 25 datos y para el Throughput se tomó 26 datos.

## **3.4. HIPOTESIS.**

**Hipótesis Nula (H<sub>0</sub>)**= El rendimiento la red del escenario1(HMIPv6) es igual al rendimiento de la red del escenario final (IntServ6 sobre HMIPv6-BI).

**Hipótesis Alternativa (H<sub>i</sub>)** = El rendimiento de la red del escenario1(HMIPv6) no es igual al rendimiento de la red del escenario final (IntServ6 sobre HMIPv6-BI).

### 3.4.1. OPERACIONALIZACIÓN DE LAS VARIABLES

<b><u>VARIABLES</u></b>	<b><u>CONCEPTO</u></b>	<b><u>INDICADORES</u></b>	<b><u>INSTRUMENTO</u></b>
<p><b><u>Variable Independiente:</u></b> Simulación de IntServ6 sobre el protocolo HIPv6-BI en redes móviles IPv6</p>	<p>Diseño de redes móviles IPv6 para la simulación de Intserv6 sobre el protocolo HMIPv6-BI</p>	<p>-Topología de red -Nodo de partida -Nodo de llegada</p>	<p>Software para análisis de redes</p>
<p><b><u>Variable Dependiente:</u></b> Calidad de servicio y rendimiento en redes móviles de comunicación IPv6</p>	<p>Optimizar la velocidad y latencia.</p>	<p>-Jitter -Throughput -VER  -Pérdida de paquetes</p>	<p>Resultados de simulación</p>

### 3.5. PROCEDIMIENTOS Y ANÁLISIS

Para mejorar el rendimiento en una red móvil que utiliza el protocolo IPV6, se planteó diseñar un escenario que permita medir parámetros para comprobar si con la implementación de IntServ6 sobre el protocolo HMIPv6-BI se logra mejorar la red, se realizó el siguiente procedimiento:

1. Buscar toda la información sobre HMIPv6-BI, IntServ6 y rendimiento de red móvil.
2. Creación de máquina virtual sobre un virtualizador VVMWARE Workstation Pro.
3. Instalación de NS3 y demás herramientas para visualizar y medir los parámetros de la red.
4. Creación de una red móvil con el protocolo HMIPv6 y medir sus parámetros.
5. Creación de una nueva red móvil con HMIPv6-BI y comparar con la anterior red de acuerdo a los indicadores.
6. Implementar IntServ6 en la red con HMIPv6-BI para verificar que exista mayor transmisión de paquetes ya que el cálculo de la clave HASH solo se hace en el destino y origen mientras que por los demás nodos viaja como cabecera en el paquete a transmitir.

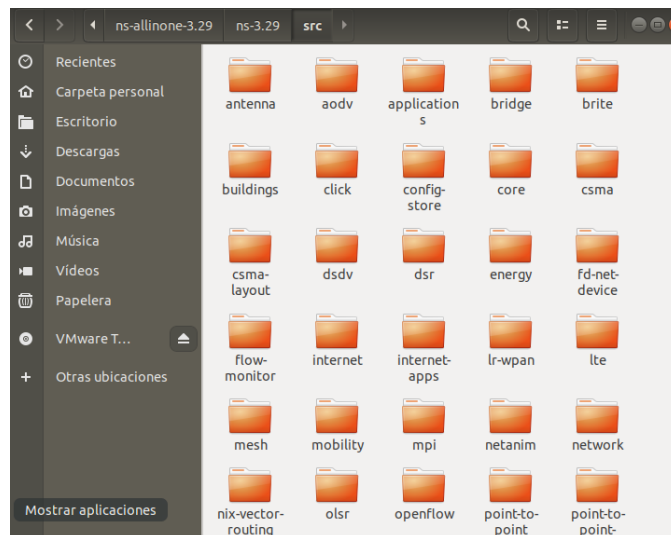
### 3.5.1. Desarrollo de la simulación en el software de simulación NS3

#### 3.5.1.1. Simulador De Redes Versión 3

EL software de simulación NS3 es un programa que permite realizar simulaciones discretas de redes ya sea ethernet, wifi, satelitales entre otras, así como la combinación de las antes mencionadas y posee características que la distinguen de los demás programas de simulación y se detalla a continuación (De Ingeniería, Electrónica, & Redes):

- El software NS3 tiene bibliotecas incluidas que se pueden combinar entre sí para el desarrollo de un programa, así como también con nuevas u otras bibliotecas externas.
- Se puede hacer el desarrollo del programa ya sea en C++ o Python.
- Tiene herramientas de visualización y análisis, como lo son NetAnim y Tracemetrics.

NS3 se divide en varios módulos como se muestra en la **Figura 13**.

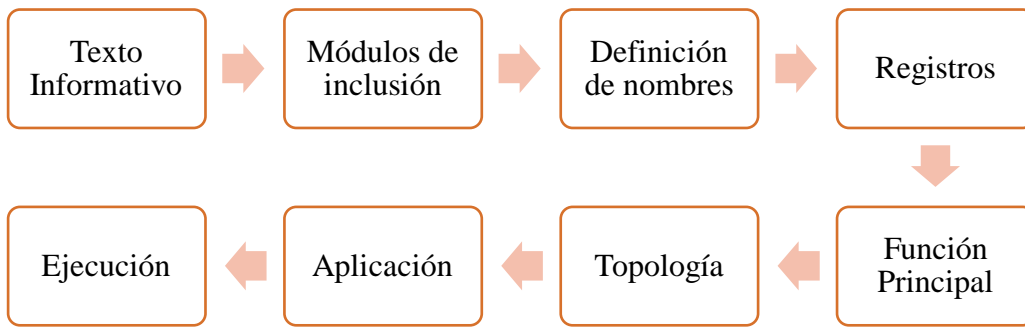


*Figura 13. Bibliotecas de NS-3*

**Fuente:** Autora

#### 3.5.1.2. Estructura de un Script en NS3 utilizada en el proyecto.

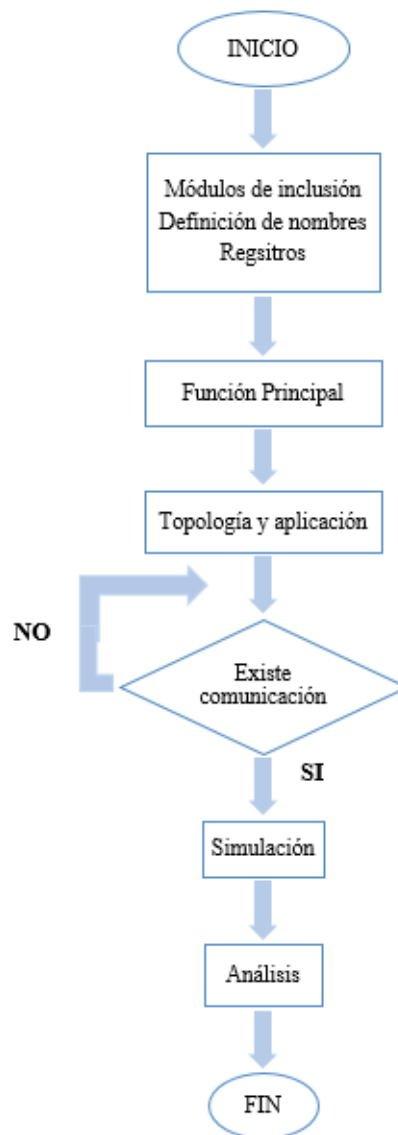
El script es un documento de texto que se puede programar en código de programación C++ o Python, para la presente investigación se enfoca en el script código C++ y se detalla las sentencias que se utilizan se muestra en la **Figura 14**.



*Figura 14. Estructura de un script*

**Fuente:** Autora

**3.5.1.3. Diagrama de Flujo para el desarrollo de la investigación.**



*Figura 15. Diagrama de flujo para el desarrollo del proyecto*

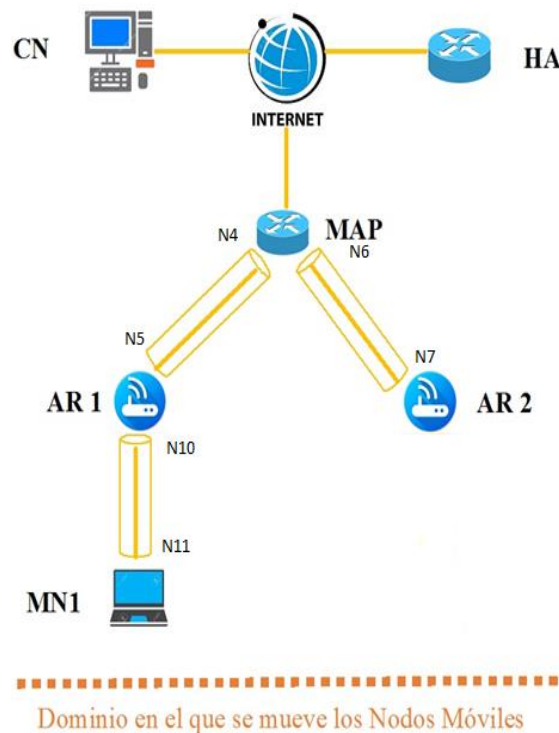
**Fuente:** Autora

Para el desarrollo de script en C++ para la creación de una red, se debe estructurar de manera que sea un proceso ordenado y adecuado para la correcta implementación como lo muestra la **Figura 15**, se realiza el proceso de inclusión de librerías registros, como la creación de la topología que se desea implementar para la posterior implementación de aplicaciones con los protocolos de movilidad como el protocolo de calidad de servicio verificando la comunicación de la red completa se procede a la simulación y extracción de datos para su análisis.

Se observa el Diagrama de flujo y en el **Anexo 1** se describe el proceso para el desarrollo del proyecto de implementación.

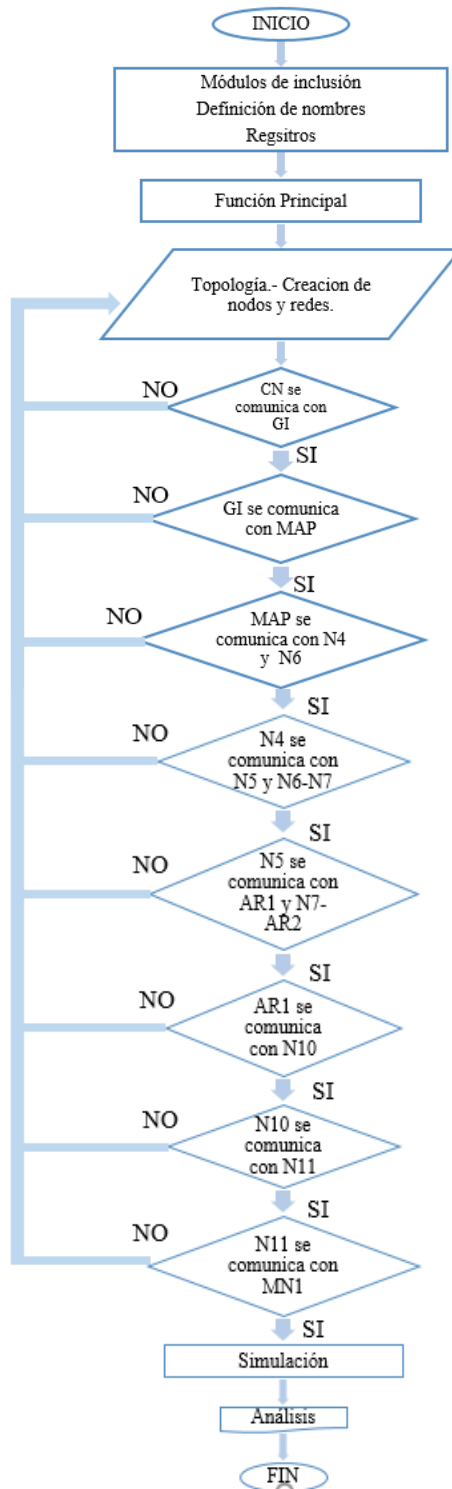
### 3.5.2. Diseño del escenario inicial

Para el desarrollo del escenario preliminar, se inició por una simulación de una red móvil que utiliza el protocolo HMIPv6, y en la **Figura 16** se puede observar la topología, mientras que en la **Figura 17** se muestra el diagrama de flujo para el desarrollo del script en NS3.



**Figura 16.** Topología para el escenario 1

**Fuente:** Autora



*Figura 17. Diagrama de flujo para implementar el escenario1.*

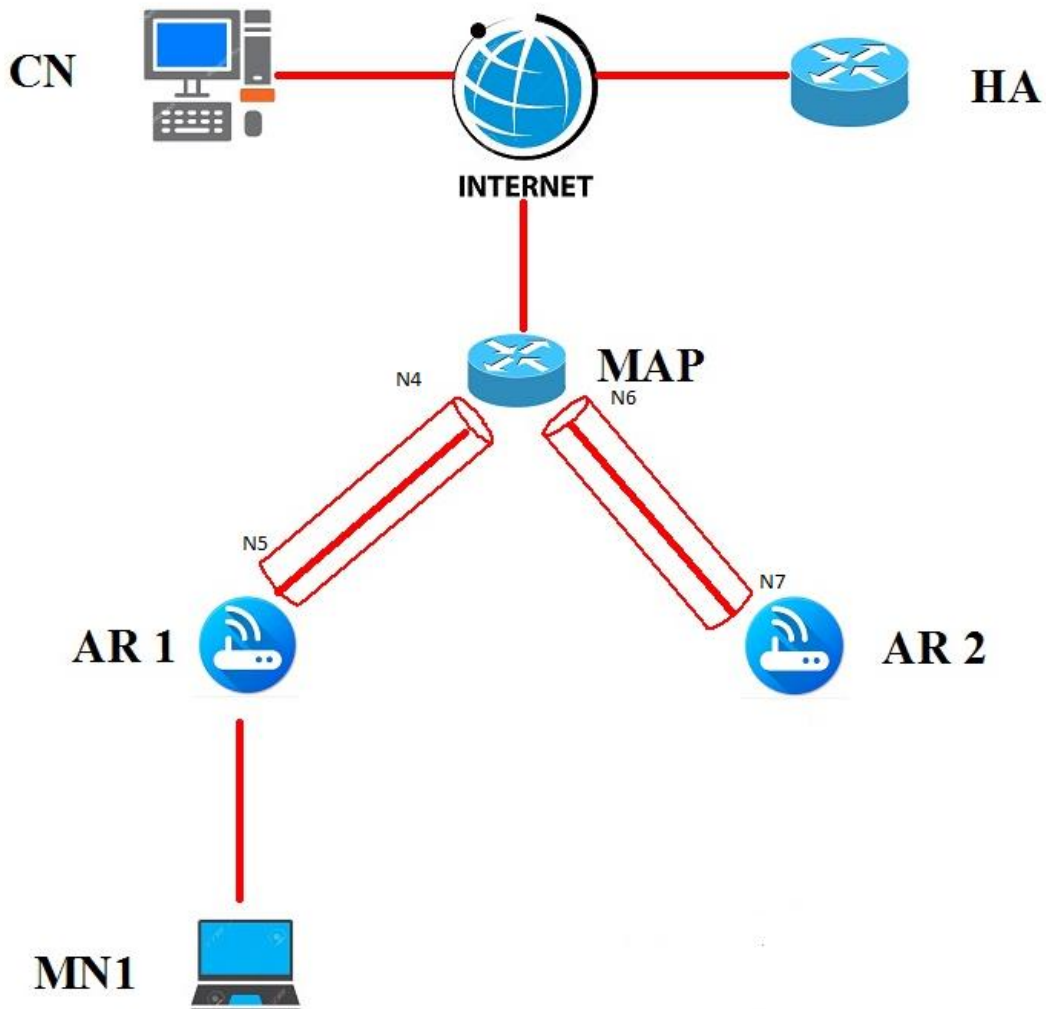
**Fuente:** Autora

Para ver en código ir al **Anexo 2.**



### 3.5.3. Diseño de la red móvil implementando HMIPv6-BI

Una vez creado el escenario1, se procede a la creación de la misma red aplicando el protocolo HMIPv6-BI que lo que realiza es quitar la encapsulación del paquete para la comunicación entre AR1 Y MN1, de acuerdo al Diagrama de flujo de la **Figura 19**, respetando la topología de la **Figura 18**. (Ver código en **Anexo 3**)

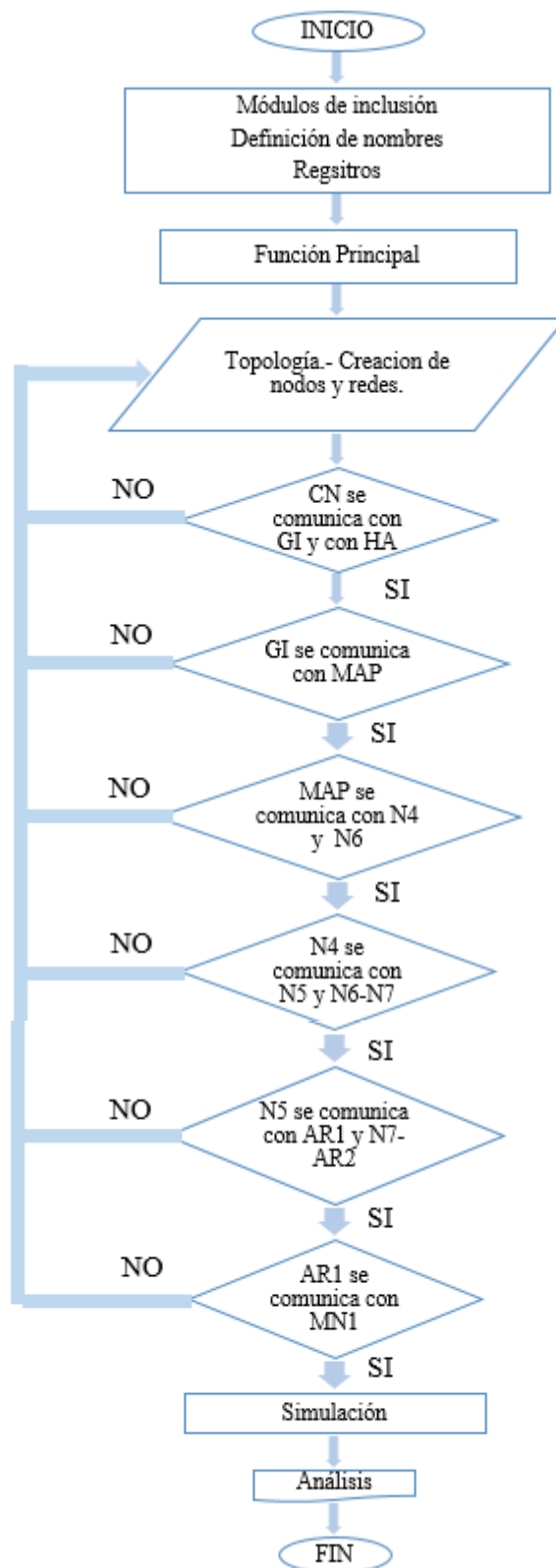


-----

**Dominio en el que se mueve los Nodos Móviles**

*Figura 18. Topología para implementar del protocolo HMIPv6-BI*

**Fuente:** Autora



*Figura 19. Diagrama de Flujo para implementar el protocolo HMIPv6-BI*

**Fuente:** Autora

### 3.5.4. Diseño de IntServ6 sobre HMIPv6-BI

Para la implementación de calidad de servicio en redes móviles, se utiliza la Arquitectura de Servicios Integrados (Padilla, Paradells, & Rodríguez) y la propuesta de IntServ6 al igual que IntServ utiliza señalización RSVP.

La idea general de la propuesta de IntServ6 es que la Arquitectura soporte calidad de servicio por flujos, ya que hace reservas de recursos para flujos antes de empezar la transmisión. Y el cálculo de la clave HASH solo se realiza en el Host Origen, y para no interferir con los demás procesos del software NS3 se crea nuevas librerías .h y .cc, los mismos que deben ser ejecutados y realizar las correcciones adecuadas hasta que funcionen de manera ideal; a continuación se describen los módulos creados en la

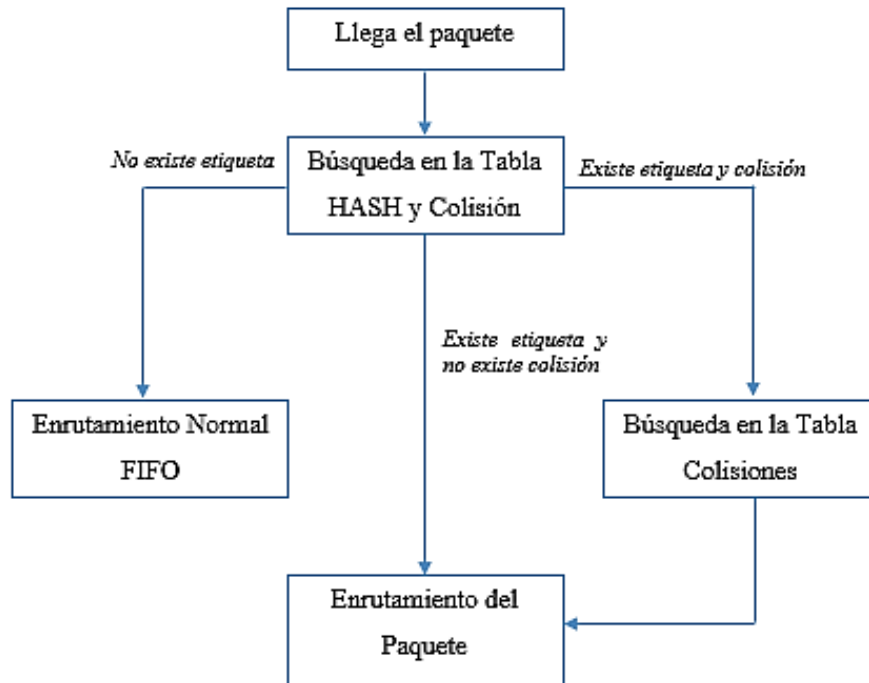
**Tabla 7:**

*Tabla 7. Descripción de módulos de inclusión para el escenario final*

Módulos de inclusión creados	Descripción
classifier-IntServ6.h	Se diferencian al clasificador hash que viene por defecto en NS3, y se creó con el fin de realizar el cálculo del clave hash con los datos de dirección IP origen y destino. Y Realizando la función XOR. Y así generar una tabla HASH y una de colisiones
ip.h	Se realiza para aumentar el número de bits de la cabecera para poder ingresar el numero HASH a la etiqueta de flujo en la cabecera del protocolo IPv6.

**Fuente:** Autora

La topología que se utiliza es la misma que la red que usa el protocolo HMIPv6-BI de la **Figura 18**. Y se adiciona la manera en que se transmite y clasifica los paquetes, ya que el cálculo de la clave HASH solo se realiza en el Host de Origen y se añade como etiqueta del flujo de paquete y pasa por todos los routers haciendo el procedimiento que se describe en la **Figura 20**.



**Figura 20.** *Funcionamiento IntServ6*

**Fuente:** Autora

El escenario final de simulación busca mejorar el rendimiento de la red en general, por lo que en el capítulo 4 se analizará los resultados y compararlos con el escenario1, para comprobar que la red móvil mejore con esta propuesta, el código que se implementó ir al **Anexo 3.**

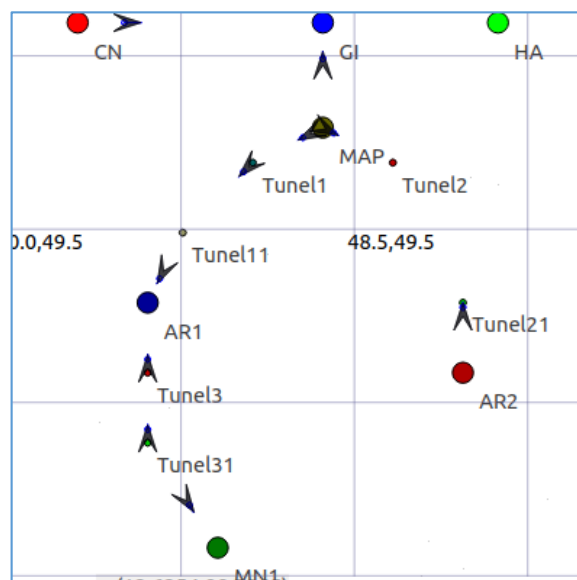
## CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

Este capítulo detalla las pruebas, simulaciones que se realizaron de los scripts creados en NS3, la visualización con la herramienta NetAnim y con el analizador Tracemetrics de una red móvil IntServ6 sobre el protocolo HMIPv6-BI para mejorar el rendimiento de la red que utiliza el protocolo IPV6.

### 4.1. SIMULACIONES

#### 4.1.1. SIMULACIÓN DE LA RED MÓVIL CON EL PROTOCOLO HMIPV6

Para la ubicación de los nodos de la red se utiliza la librería de NetAnim para poder dar coordenadas, color y tamaño de cada nodo de la red como se muestra en la **Figura 21**, la comunicación entre cada nodo de la red con el protocolo ICMPv6 con un tamaño de paquete de 1500 bytes, intervalo entre paquetes de 2ms, con una velocidad de 11Mpbs de un enlace inalámbrico y una arquitectura sencilla.



**Figura 21.** Comunicación entre CN y MN1 del escenario1

**Fuente:** Autora

#### 4.1.1.1. *Transmisión de paquetes y tiempos del escenario1.*

Los pasos para establecer la comunicación entre CN y MN1 son los siguientes.

1. **Comunicación entre el nodo HA y CN.** En nodo Home Agent sobre el cual se realiza el enlace local del nodo móvil MN1, se comunica con CN (Nodo correspondiente), para registrar todas las direcciones posibles existentes en el nodo HA.
2. **Comunicación entre el nodo CN y GI.-** El nodo correspondiente CN que es un nodo estático o también puede ser móvil, busca comunicarse con el nodo MN1, debe enviar un paquete a GI (internet) que pedirá la tabla de direcciones actualizadas conocido como *Binding Cache*.
3. **Comunicación entre el nodo GI y MAP.** - El paquete sale del nodo internet GI y llega al nodo MAP, que es el que punto de anclaje móvil. Que es el que direcciona hacia los routers de acceso AR1 y AR2 en busca del MN1.
4. **Comunicación entre el nodo MAP - AR1 y AR2.-** Del nodo MAP salen paquetes hacia el nodo AR1 y AR2, pero en este caso para la simulación se tomó en referencia el RFC2473 indica que para la tunelización existe un nodo punto de entrada del túnel en este caso son TUNEL1 y TUNEL2. Y el otro nodo punto de salida del túnel que son TUNEL11 y TUNEL21 que se comunican con AR1 y AR2 respectivamente.
5. **Comunicación entre el nodo AR1 y MN1.** En este punto el paquete es encapsulado hacia el MN1, con la nueva nLCoA y que igual ocupa nodo ingreso de túnel (TUNEL3) y el nodo salida de túnel (TUNEL31) según el RFC2473

En la **Tabla 8** se muestra los datos obtenidos en la transmisión entre los nodos CN y MN1 bidireccional de acuerdo al **Anexo 4** se detallan tiempos de acuerdo a los datos obtenidos en NetAnim. El tiempo de conexión entre CN-MN1 de 0,025424ms obtenidos en el simulador se observa el dato de color rojo, mientras que para la gestión de movilidad que es la comunicación entre el MAP es el dato de color azul 0,01631 ms.

**Tabla 8.** *Tiempos de conexión entre CN-MN1*

<b>CONEXIÓN</b>	<b>TIEMPO</b>	<b>UNIDAD</b>
CN-GI	0,00415	ms
HA-GI	0,003343	ms
GI-MAP	0,001	ms
MAP-TUNEL1	0,006797	ms
TUNEL1-TUNEL11	0,002	ms
TUNEL11-AR1	0,001	ms
AR1-TUNEL3	0	ms
TUNEL3-TUNEL31	0,001	ms
TUNEL31-MN1	0,006134	ms
<b><i>TX:CN-MN1</i></b>	<b><i>0,025424</i></b>	<b><i>ms</i></b>
<b><i>TX:MAP-MN1</i></b>	<b><i>0,016931</i></b>	<b><i>ms</i></b>

**Fuente:** Autora

#### **4.1.1.2. Medición de Throughput del escenario1**

Para el análisis mediante Tracemetrics se debe crear archivos .tr, también se crean archivos. pcap para analizar mediante Wireshark y se genera las siguientes tablas con la información obtenida de los archivos, donde muestra a más detalles el envío de mensajes de un nodo a otro, mientras que Tracemetrics permite guardar los valores medidos del Throughput en B/s pero la medida del rendimiento se lo realiza en Mbps por lo que el resultado obtenido en Tracemetrics se debe multiplicar \*8 bits de tiene cada byte. La **Tabla 9**, muestra el promedio de los mensajes que fueron recibidos de manera exitosa por unidad de tiempo.

**Tabla 9.** *Medición del Throughput del escenario1*

<b>NODOS</b>	<b>Throughput (Mbps)</b>	<b>Goodput (Mbps)</b>
HA	0,632961468	0,632961468
CN	0,632961468	0,632961468
GI	1,265922937	1,265922937
MAP	0,632961468	0,632961468
TUNEL1	0,632961468	0,632961468
TUNEL11	0,632961468	0,632961468
AR1	1,265922937	1,265922937
TUNEL3	0,632961468	0,632961468
TUNEL31	0,632961468	0,632961468
MN1	0,632961468	0,632961468
<b>Promedio</b>	<b>0,759553762</b>	<b>0,759553762</b>

**Fuente:** Autora

El Throughput es el rendimiento de la red y equivale a 0,759553762 Mbps, mientras que el Goodput es la información útil y es igual al Throughput de la red.

#### 4.1.1.3. Medición del BER del escenario1

El software NS3 no permite medir directamente el BIT RATE ERROR, pero mediante la fórmula de la **Figura 22**, indica una distribución de Bernoulli con un intervalo de confianza del 99.8% se puede observar en la **Tabla 10**.

$$BER = \frac{\text{Número de paquetes errados} * (1 - 99,8 \%)}{\text{Número de paquetes transmitidos}}$$

**Figura 22.** Fórmula para calcular el BER

**Fuente:** (Hugo et al., n.d.)

De acuerdo a la información de la cantidad de paquetes errados y enviados se obtiene la **Tabla 10** y se observa un BER de 0,000211.

**Tabla 10.** BER del escenario1

ESCENARIO	PAQUETES ENVIADOS	PAQUETES ERRADOS	BER
Red con HMIPv6	57	6	2,1053E-04

**Fuente:** Autora

#### 4.1.1.4. Medición del Jitter del escenario1.

Mediante la herramienta Flow Monitor se obtiene JitterSum que es la suma total del Jitter de extremo a extremo, y posterior se procede a realizar la suma y promedio del Jitter en ms como se muestra en la **Tabla 11**.

**Tabla 11.** JITTER del escenario1

CONEXIÓN	Dirección de Origen	Dirección Destino	Jitter(ms)
HA-CN	2001:2::200:ff:fe00:1	2001:::200:ff:fe00:1	9,234
CN-GI	2001:::200:ff:fe00:1	2001:1::200:ff:fe00:1	2,178
GI-MAP	2001:1::200:ff:fe00:1	2001:3::200:ff:fe00:1	0
MAP-TUNEL1	2001:3::200:ff:fe00:1	2001:4::200:ff:fe00:1	4,768
TUNEL1-TUNEL11	2001:4::200:ff:fe00:1	2001:5::200:ff:fe00:1	3,879
TUNEL11-AR1	2001:5::200:ff:fe00:1	2001:8::200:ff:fe00:1	8,0023
AR1-TUNEL3	2001:8::200:ff:fe00:1	2001:10::200:ff:fe00:1	0
TUNEL3-TUNEL31	2001:10::200:ff:fe00:1	2001:11::200:ff:fe00:1	9,765
TUNEL31-MN1	2001:11::200:ff:fe00:1	2001:12::200:ff:fe00:1	12,98
<b>PROMEDIO</b>			5,645
<b>JITTER TOTAL</b>			50,8063

**Fuente:** Autora



#### 4.1.1.5. Pérdida de paquetes del escenario 1.

Mediante los parámetros que se muestra con el archivo.xml en Flow Monitor, se determina la pérdida de paquetes en porcentaje, el cual permite obtener la información de pérdida total de paquetes, así como promedio como lo muestra la siguiente **Tabla 12**.

*Tabla 12. Pérdida de paquetes del escenario 1*

CONEXIÓN	Dirección de Origen	Dirección Destino	Pérdida de paquetes %
HA-CN	2001:2::200:ff:fe00:1	2001:::200:ff:fe00:1	0,002
CN-GI	2001:::200:ff:fe00:1	2001:1::200:ff:fe00:1	0
GI-MAP	2001:1::200:ff:fe00:1	2001:3::200:ff:fe00:1	0,001
MAP-TUNEL1	2001:3::200:ff:fe00:1	2001:4::200:ff:fe00:1	0,02
TUNEL1-TUNEL11	2001:4::200:ff:fe00:1	2001:5::200:ff:fe00:1	0,1
TUNEL11-AR1	2001:5::200:ff:fe00:1	2001:8::200:ff:fe00:1	0
AR1-TUNEL3	2001:8::200:ff:fe00:1	2001:10::200:ff:fe00:1	0,001
TUNEL3-TUNEL31	2001:10::200:ff:fe00:1	2001:11::200:ff:fe00:1	0,01
TUNEL31-MN1	2001:11::200:ff:fe00:1	2001:12::200:ff:fe00:1	0,002
<b>PROMEDIO</b>			<b>0,015</b>
<b>PÉRDIDA DE PAQUETES TOTAL</b>			<b>0,136</b>

Fuente: Autora

#### 4.1.2. SIMULACIÓN DE LA RED MÓVIL CON INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI

Para la comunicación entre los nodos de la red móvil se siguen los pasos a continuación.

- 1. Comunicación entre HA y CN.** - En nodo HA, que es el Home Agent sobre el cual se realiza el enlace local del nodo móvil MN1, se comunica con CN (Nodo correspondiente), para registrar todas las direcciones posibles existentes en el nodo HA.
- 2. Comunicación entre el nodo CN y GI.** - El nodo correspondiente CN que es un nodo estático o también puede ser móvil, busca comunicarse con el nodo MN1, debe enviar un paquete a GI(internet) que pedirá la tabla de direcciones actualizadas conocido como Binding Cache
- 3. Comunicación entre el nodo GI y MAP.** -El paquete sale del nodo internet GI y llega al nodo MAP, que es el que punto de anclaje móvil. Que es el que direcciona hacia los routers de acceso AR1 y AR2 en busca del MN1.
- 4. Comunicación entre el nodo MAP y AR1 y AR2.** -Del nodo MAP salen paquetes hacia el nodo AR1 y AR2, pero en este caso para la simulación se tomó

en referencia el RFC2473 que describe la tunelización en un nodo punto de entrada del túnel en este caso son TUNEL1 y TUNEL2, y el otro nodo punto de salida del túnel que son TUNEL11 y TUNEL21 que se comunican con AR1 y AR2 respectivamente.

5. **Comunicación entre el nodo AR1 y MN1.** -En este punto el paquete es encapsulado hacia el MN1, con la nueva ubicación nLCoA(New Local Care-of Adres).

#### 4.1.2.1. *Tiempo de transmisión entre CN-MN1 del escenario final.*

De acuerdo al **Anexo 5** se generó la **Tabla 13**, se especifica la comunicación bidireccional entre el nodo correspondiente(CN) y el nodo móvil(MN1).

**Tabla 13.** *Tiempos de conexión bidireccional entre CN-MN1*

<b>CONEXIÓN</b>	<b>TIEMPO</b>	<b>UNIDAD</b>
CN-GI	0,002	ms
HA-GI	0,003343	ms
GI-MAP	0,005	ms
MAP-TUNEL1	0,004498	ms
TUNEL1-TUNEL11	0,002	ms
TUNEL11-AR1	0,006	ms
AR1-MN1	0,00634	ms
<b><i>TX:CN-MN1</i></b>	<b><i>0,029181</i></b>	<b><i>ms</i></b>
<b><i>TX:MAP-MN1</i></b>	<b><i>0,018838</i></b>	<b><i>ms</i></b>

**Fuente:** Autora

El resultado en azul muestra el tiempo de gestión de movilidad entre el MAP y MN1, mientras que 0,029181ms valor de color rojo muestra el resultado de la comunicación entre CN-MN1.

#### 4.1.2.2. *Medición de Throughput del escenario final.*

Para obtener el rendimiento de la red del script del escenario final, se implementó IntServ6 sobre HMIPV6-BI, mediante la herramienta Tracemetrics se determinó un Throughput promedio de la red de 1,184682061Mbps que es igual al Goodput (información útil) y se detalla con mayor precisión el rendimiento de cada nodo implementado en la **Tabla 14**.

**Tabla 14. Throughput del escenario final**

<b>NODOS</b>	<b>Throughput (Mbps)</b>	<b>Goodput (Mbps)</b>
HA	0,631830433	0,631830433
CN	0,631830433	0,631830433
GI	1,895491298	1,895491298
MAP	1,895491298	1,895491298
TUNEL1	1,263660865	1,263660865
TUNEL11	1,263660865	1,263660865
AR1	1,263660865	1,263660865
MN1	0,631830433	0,631830433
<b>Promedio</b>	<b>1,184682061</b>	<b>1,184682061</b>

**Fuente:** Autora

#### **4.1.2.3. Medición del BER del escenario final.**

Para el cálculo del BER (Bit Rate Error) se realiza mediante la ecuación descrita en la sección 4.2.1.3. que sirve para determinar el número de bits errados por unidad de tiempo de acuerdo al paquete enviados y errados. Ver **Tabla 15.**

**Tabla 15. BER del escenario final**

<b>ESCENARIO</b>	<b>PAQUETES ENVIADOS</b>	<b>PAQUETES ERRADOS</b>	<b>BER</b>
Red con IntServ6 sobre HMIPv6-BI	72	3	8,3333E-05

**Fuente:** Autora

#### **4.1.2.4. Medición del Jitter del escenario final.**

Para conocer el Jitter promedio o fluctuación de retardo como también es llamada, de acuerdo a lo datos visualizados en la herramienta Flow Monitor se realizó la **Tabla 16** se detalla el Jitter de cada conexión, la suma total y el promedio.

**Tabla 16. Jitter del escenario final**

CONEXIÓN	Dirección de Origen	Dirección Destino	Jitter (ms)
HA-CN	2001:1::200:ff:fe00:1	2001:::200:ff:fe00:1	2,891
CN-GI	2001:::200:ff:fe00:1	2001:2::200:ff:fe00:1	2,003
GI-MAP	2001:2::200:ff:fe00:1	2001:3::200:ff:fe00:1	0
MAP-TUNEL1	2001:3::200:ff:fe00:1	2001:4::200:ff:fe00:1	1,876
TUNEL1-TUNEL11	2001:4::200:ff:fe00:1	2001:6::200:ff:fe00:1	3,467
TUNEL11-AR1	2001:6::200:ff:fe00:1	2001:8::200:ff:fe00:1	6,56
AR1-MN1	2001:8::200:ff:fe00:1	2001:10::200:ff:fe00:1	3
<b>PROMEDIO</b>			<b>2,828</b>
<b>JITTER TOTAL</b>			<b>19,797</b>

Fuente: Autora

#### 4.1.2.5. Pérdida de paquetes del escenario final.

La pérdida de paquetes se define como el porcentaje de paquetes que no logra llegar a su destino y es recomendable que se menor a 2.5%. De acuerdo a los datos obtenidos en Flow Monitor se realizó la **Tabla 17** que detalla la pérdida de paquetes de cada conexión, promedio y la suma total.

**Tabla 17. Pérdida de paquetes del escenario final**

CONEXIÓN	Dirección de Origen	Dirección Destino	Pérdida de paquetes %
HA-CN	2001:1::200:ff:fe00:1	2001:::200:ff:fe00:1	0,003
CN-GI	2001:::200:ff:fe00:1	2001:2::200:ff:fe00:1	0,001
GI-MAP	2001:2::200:ff:fe00:1	2001:3::200:ff:fe00:1	0
MAP-TUNEL1	2001:3::200:ff:fe00:1	2001:4::200:ff:fe00:1	0,001
TUNEL1-TUNEL11	2001:4::200:ff:fe00:1	2001:6::200:ff:fe00:1	0,001
TUNEL11-AR1	2001:6::200:ff:fe00:1	2001:8::200:ff:fe00:1	0,001
AR1-MN1	2001:8::200:ff:fe00:1	2001:10::200:ff:fe00:1	0,002
<b>PROMEDIO</b>			<b>0,001</b>
<b>PÉRDIDA DE PAQUETES TOTAL</b>			<b>0,009</b>

Fuente: Autora

## 4.2. ANÁLISIS

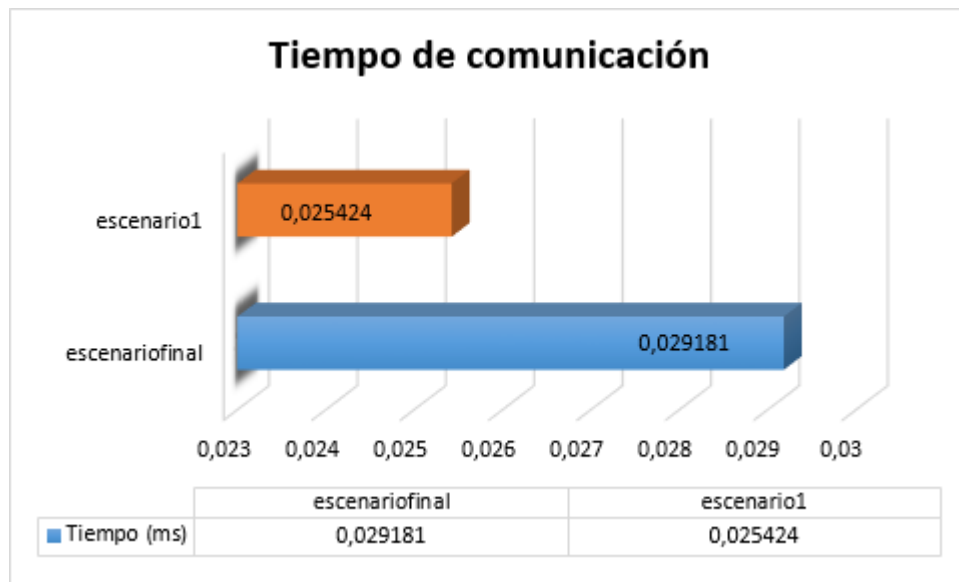
### 4.2.1. Análisis de tiempos de comunicación.

De acuerdo a la información obtenida en la **Tabla 8** de los tiempos del escenario1 y la **Tabla 13** que es el escenario final, se realizó la **Tabla 18**, que hace referencia a una comparativa entre los tiempos de conexión y como se puede observar la transmisión de paquetes entre CN-MN1 no se mejora en el escenario final ya que el escenario1 tiene 0,003757 ms menos para establecer la comunicación, debido a que HMIPv6-BI con IntServ6 se tiene que verificar con las tablas de reserva para comprobar la clave HASH que viene como etiqueta de flujo en la cabecera del paquete transmitido como se muestra en la **Figura 23**.

**Tabla 18.** Comparación de los tiempos de conexión entre escenario1 y escenario final

	ESCENARIO1	ESCENARIOFINAL
<i>TX:CN-MNI</i>	0,025424 ms	0,029181 ms
<i>TX:MAP-MNI</i>	0,016931 ms	0,018838 ms

**Fuente:** Autora



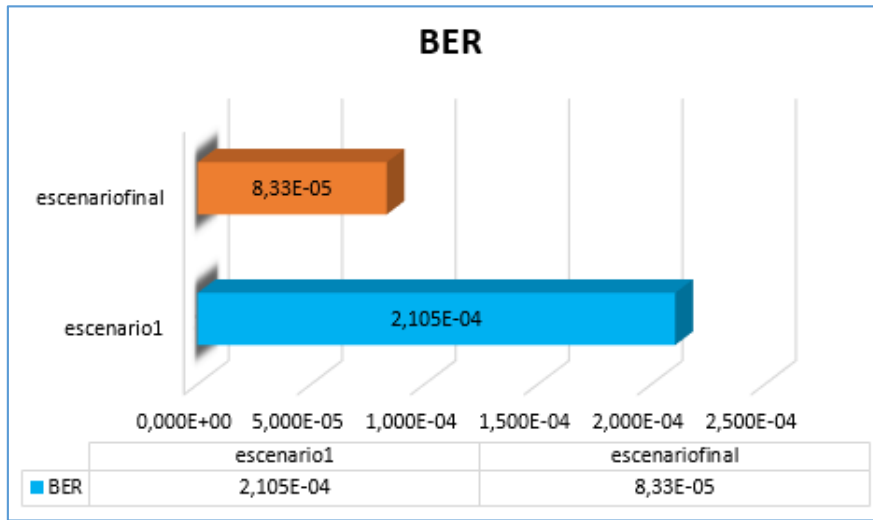
**Figura 23.** Comparación de tiempo de comunicación entre el escenario1 y escenario final

**Fuente:** Autora

### 4.2.2. Análisis del BER.

Para el análisis del Bit Rate Error se realizó una comparativa con los valores de la sección 4.2.1.3 y 4.2.2.3 del escenario 1 y escenario final respectivamente, mediante lo cual se

determina que la tasa del BER es mayor en el escenario 1 que en el escenario final. Por lo tanto, se determinó que con la implementación de IntServ6 sobre HMIPv6-BI mejora la cantidad de bit errados en la red como se puede observar en la **Figura 24**.

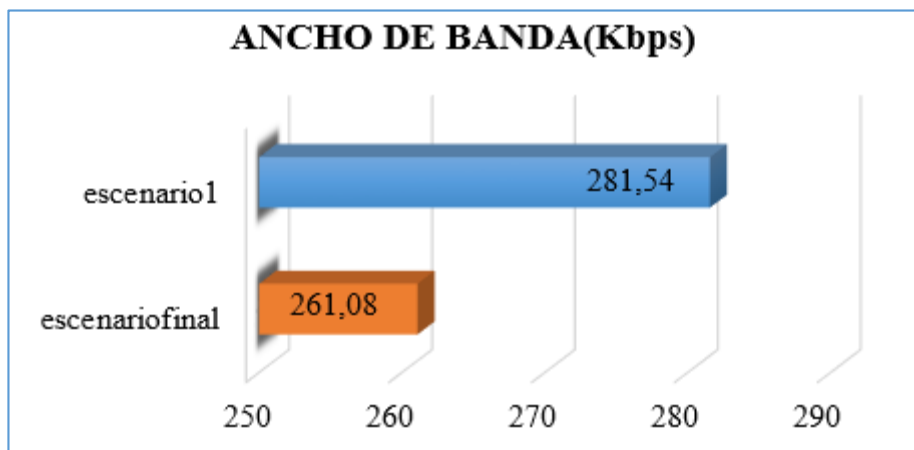


*Figura 24. Comparación del BER entre el escenario1 y escenario final*

**Fuente:** Autora

#### 4.2.3. Análisis del Ancho de Banda

En la **Figura 25** se muestra los resultados de la simulación en NS-3 para el ancho de banda en el canal de radio que es desde el AR1 hasta MN1 con 11Mbps que es de un enlace inalámbrico, un intervalo entre paquetes de 0,5s, tamaño de paquetes de 1400 bytes con la arquitectura sencilla de los escenario1 y del escenario final.

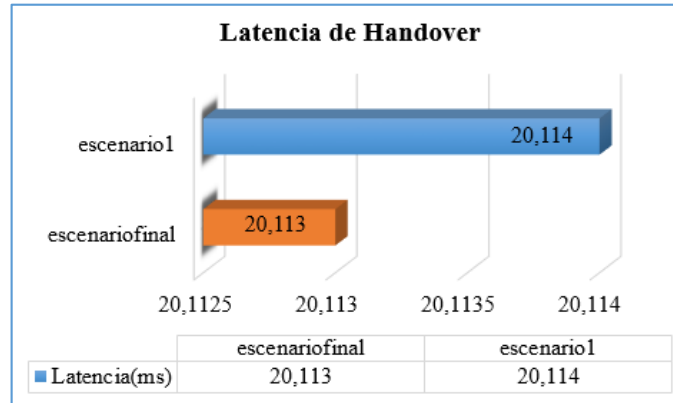


*Figura 25. Comparación del Ancho de Banda entre el escenario1 y escenario final*

**Fuente:** Autora

#### 4.2.4. Análisis de Latencia de Handover

Para el análisis de latencia de handover se analiza en handover local que es el traslado del MN1 en la misma subred en el dominio del MAP, según (Becerra, Padilla, & Paradells, 2011) el protocolo HMIPv6 (escenario1) utiliza Fast Handover que se caracteriza por la reducción de la latencia y HMIPv6-BI (escenario final) que modifica o recorta el túnel que va desde AR1 a MN1, pero no afecta de manera significativa a los valores de latencia de handover sin tener en cuenta los retardos como se muestra en la **Figura 26**.



**Figura 26.** Comparación de latencia de handover entre el escenario1 y escenario final

**Fuente:** Autora

### 4.3. ANÁLISIS ESTADÍSTICO

Para realizar el análisis de manera estadística como primer punto es determinar si existe una hipótesis que demostrar y en el presente proyecto de investigación existen hipótesis para verificar si la red del escenario final mejora en comparación con la del escenario1. Por lo cual el análisis se realiza de acuerdo a shapiro-wilk que se utiliza cuando existen menos de 50 muestras obtenidas de las simulaciones en el software NS3, y se procede a realizar la prueba de normalidad que es necesario para la futura utilización de estadística paramétrica o no paramétrica.

En la **Figura 27** no todos los datos corresponden a un Pvalor menor a 0,05 por lo que se procede en estadística no paramétrica para validar el mejor rendimiento en la red del escenario final con respecto al escenario1.

**Pruebas de normalidad**

	Kolmogorov-Smirnov <sup>a</sup>			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Jitter escenariofinal(ms)	,276	15	,003	,754	15	,001
Perdida de paquetes escenario1(%)	,477	15	,000	,517	15	,000
Perdida de paquetes escenariofinal(%)	,388	15	,000	,736	15	,001
Jitter ecenario1 (ms)	,222	15	,045	,862	15	,026
Tiempo de comunicación escenariofinal(ms)	,210	15	,075	,869	15	,033
Tiempo de comunicación ecenario1 (ms)	,168	15	,200 <sup>*</sup>	,880	15	,047
Rendimiento escenario1 (B)	,485	15	,000	,499	15	,000
Rendimiento escenariofinal (B)	,249	15	,013	,806	15	,004

**Figura 27. Prueba de normalidad con IBM SPSS**

**Fuente:** Autora

Para el análisis estadístico no paramétrico se debe tener en cuenta dos tipos de hipótesis que son:

- **Hipótesis Nula:** Que es una igualdad entre las variables a comparar.
- **Hipótesis alternativa:** Es una desigualdad de las variables analizadas

Se rechaza o se aprueba una hipótesis de acuerdo al valor de significancia o conocido como P valor que es una comparativa con un 0,05 que da un 95% de confianza, y cuando  $Pvalor > 0,05$  se aprueba la hipótesis nula, mientras que  $Pvalor < 0,05$  se aprueba la hipótesis alternativa. Dicha comparación se hace de acuerdo con la mediana de todos los valores obtenidos en las simulaciones de NS-3.

De acuerdo a los datos tomados en las simulaciones se procede a ingresar los datos en el software estadístico IBM SPSS, que es el que permite desarrollar y validar o rechazar las hipótesis creadas en la investigación. Mediante IBM SPSS es posible obtener datos como mediana, media, varianza entre otros que se muestra en la **Figura 28**.



	Estadístico	Error estándar
Media	,00334167	,000405849
95% de intervalo de confianza para la media	Límite inferior	,00250743
	Límite superior	,00417590
Media recortada al 5%	,00336663	
Mediana	,00334300	
Varianza	,000	
Desviación estándar	,002108855	
Mínimo	,000100	
Máximo	,006134	
Rango	,006034	
Rango intercuartil	,003000	
Asimetría	-,067	,448
Curtosis	-1,417	,872
Media	,07243148	,047938157
95% de intervalo de confianza para la media	Límite inferior	-,02610681
	Límite superior	,17096977
Media recortada al 5%	,02729095	
Mediana	,00334300	
Varianza	,062	
Desviación estándar	,249093971	
Mínimo	,000000	
Máximo	,980980	
Rango	,980980	
Rango intercuartil	,005797	
Asimetría	3,462	,448
Curtosis	10,828	,872

**Figura 28.** Datos descriptivos de tiempos de conexión con IBM SPSS

**Fuente:** Autora

Para el análisis descriptivo se realiza dos hipótesis que posteriormente son validadas o rechazadas mediante el mismo programa IBM SPSS.

#### **4.3.1. Análisis estadístico de Throughput**

El análisis descriptivo del Throughput se establece dos hipótesis que posteriormente son validadas o rechazadas mediante el mismo programa IBM SPSS.

**Hipótesis Nula (Ho)**= El Throughput de la red del escenario1(HMIPV6) es igual al Throughput del escenario final (IntServ6 sobre HMIPV6-BI).

$$H_o: \mu_{ThrEsc1} = \mu_{ThrEsc2}$$

**Hipótesis Alternativa (Hi)** = El Throughput de la red del escenario1(HMIPV6) no es igual al Throughput de la red del escenario final (IntServ6 sobre HMIPV6-BI).

$$H_i : \mu_{ThrEsc1} \neq \mu_{ThrEsc2}$$

#### Resumen de contrastes de hipótesis

	Hipótesis nula	Prueba	Sig.	Decisión
1	La mediana de las diferencias entre Rendimiento escenario1 (B) y Rendimiento escenariofinal (B) es igual a 0.	Prueba de rangos con signo de Wilcoxon para muestras relacionadas	,038	Rechace la hipótesis nula.

Se muestran significaciones asintóticas. El nivel de significación es ,05.

*Figura 29. Contraste hipótesis de Throughput con IBM SPSS*

**Fuente:** Autora

De acuerdo a la **Figura 29** se rechaza la hipótesis nula, y se acepta la hipótesis alternativa citando que los rendimientos de los dos escenarios no son iguales, que es uno de los parámetros que se mejora en la simulación en NS-3.

#### 4.3.2. Análisis estadístico de Jitter

Para el análisis descriptivo se realiza dos hipótesis que posteriormente son validadas o rechazadas mediante el mismo programa IBM SPSS.

**Hipótesis Nula (Ho)**= El jitter de la red del escenario1(HMIPV6) es igual al jitter de la red del escenario final (IntServ6 sobre HMIPV6-BI).

$$H_o : \mu_{JitterEsc1} = \mu_{JitterEsc2}$$

**Hipótesis Alternativa (Hi)** = El jitter de la red del escenario1(HMIPV6) no es igual al jitter de la red del escenario final (IntServ6 sobre HMIPV6-BI).

$$H_i : \mu_{JitterEsc1} \neq \mu_{JitterEsc2}$$

**Resumen de contrastes de hipótesis**

	Hipótesis nula	Prueba	Sig.	Decisión
1	La mediana de las diferencias entre Jitter escenario1 (ms) y Jitter escenariofinal(ms) es igual a 0.	Prueba de rangos con signo de Wilcoxon para muestras relacionadas	,001	Rechace la hipótesis nula.

Se muestran significaciones asintóticas. El nivel de significación es ,05.

*Figura 30. Contraste hipótesis de Throughput con IBM SPSS*

**Fuente:** Autora

El software IBM SPSS determino que se rechaza la hipótesis nula, por lo que el jitter de los 2 escenarios no son iguales de acuerdo al análisis estadístico no paramétrico, ver **Figura 30.**

**4.3.3. Análisis estadístico de pérdida de paquetes.**

Se realiza dos hipótesis para el análisis descriptivo que posteriormente son validadas o rechazadas mediante el mismo programa IBM SPSS.

**Hipótesis Nula (Ho)**= La pérdida de paquetes de la red del escenario1(HMIPV6) es igual a la pérdida de paquetes de la red del escenario final (IntServ6 sobre HMIPV6-BI).

$$H_0: \mu_{\text{perdidapaquetesEsc1}} = \mu_{\text{perdidapaquetesEsc2}}$$

**Hipótesis Alternativa (Hi)** = La pérdida de paquetes de la red del escenario1(HMIPV6) no es igual a la pérdida de paquetes de la red del escenario final (IntServ6 sobre HMIPV6-BI).

$$H_i : \mu_{\text{perdidapaquetesEsc1}} \neq \mu_{\text{perdidapaquetesEsc2}}$$

**Resumen de contrastes de hipótesis**

	Hipótesis nula	Prueba	Sig.	Decisión
1	La mediana de las diferencias entre Perdida de paquetes escenario1(%) y Perdida de paquetes escenariofinal(%) es igual a 0.	Prueba de rangos con signo de Wilcoxon para muestras relacionadas	,185	Conserve la hipótesis nula.

Se muestran significaciones asintóticas. El nivel de significación es ,05.

*Figura 31. Contraste hipótesis de Throughput con IBM SPSS*

**Fuente:** Autora

En este caso se establece aprobar la hipótesis nula es decir que la pérdida de paquetes es igual o es probable que sean iguales o se diferencien en un muy poco porcentaje como se demuestra en la **Figura 31**.

#### **4.4. DISCUSIÓN**

El software de redes NS-3, es un simulador del cual no existe mucha información en la web sobre trabajos realizados en esta plataforma, por lo que el trabajo de investigación se llevó a cabo con la búsqueda de la página oficial de NS-3, como con los ejemplos de IPV6 que explica pocas redes analizando los módulos que ocupa este protocolo y de los cuales se realiza el previo análisis para entender su funcionamiento como también medir los parámetros de red mediante herramientas adicionales que permiten realizar el análisis de la red como lo son Tracemetrics, Wireshark, NetAnim y Flow Monitor.

Las herramientas que proporcionan los datos de la red necesitan ser activados y crear archivos .tar que permite el análisis en Tracemetrics y Wireshark; mientras que para el análisis de NetAnim y Flow Monitor se debe crear archivos .xml, estos archivos se crean de manera automática al ejecutar el script .cc en el terminal. Y para poder ocupar dichas herramientas debes ser instaladas de manera individual para uso posterior en el uso del escenario que se aplica el protocolo HMIPV6 como el escenario final que se implementa IntServ6 sobre HMIPV6-BI para dar calidad de servicio a la red inicial.

HMIPV6 es el protocolo que sirve para gestionar la movilidad en redes móviles que utilizan el protocolo IPV6, y realiza un túnel bidireccional desde el MAP-AR1-MN1 como se describe en el RFC2473; mientras que HMIPV6-BI recorta el túnel que solo va desde el MAP-AR1.

Y posteriormente se implementó IntServ6 sobre HMIPV6-BI para dar calidad de servicio y mediante parámetros de la red validar la propuesta de investigación de este proyecto.

## CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

### 5.1.CONCLUSIONES

Para el desarrollo del escenario preliminar se creó un script denominado escenario1.cc (protocolo HMIPv6) basado en los ejemplos existentes en NS-3, y se realizó un análisis del rendimiento de la red de acuerdo a los parámetros que se puede medir con las herramientas instalados previamente como lo son Tracemetrics, Wireshark y NetAnim-Flow Monitor. De los datos obtenidos se encontró un Throughput promedio de 0,7599 Mbps, un Jitter de 4,854ms, pérdida de paquetes de 0,0015%, un 0,00021053 de bit errados (BER), como también el ancho de banda en el medio inalámbrico entre el router de acceso(AR1) y el nodo móvil(MN1) pasando la información por la tunelización: los túneles de entrada y salida, dado como resultado de la simulación un ancho de banda de 281,54Kbps.

Al implementar HMIPv6-BI (túnel recortado) se realizó la simulación que determinó un ancho de banda en el canal de radio de 261,08Kbps y una latencia de handover de 20,113ms, se comprueba que la propuesta si ahorra el ancho de banda a comparación de la red que solo usa el protocolo HMIPv6 con un 19,46Mbps mientras que con respecto a la latencia de handover no existe una diferencia significativa ya que solo se evaluó el handover local es decir movilidad en la misma subred del punto de anclaje móvil (MAP).

Para brindar calidad de servicio se implementa IntServ6 que realiza el cálculo de clave HASH en el host origen y lo agrega en la cabecera IPV6, y permite realizar modificaciones hasta obtener la red final.

Se comprobó y valido el funcionamiento de IntServ6 sobre HMIPv6-BI mediante los parámetros de red Throughput promedio de 1,1846 Mbps, un Jitter de 1,82ms, pérdida de paquetes de 0,001%, un BER de 0,0000833; los mismos que mejoran con respecto a la red del escenario inicial que solo ocupa el protocolo HMIPv6.

Hoy en día se está estudiando el soporte para la calidad de servicio en internet, para mejorar el servicio ya existente Best-Effort con la implementación de IntServ6 sobre la red creada que utiliza el protocolo HMIPv6-BI permite comprobar el soporte de la red

con la Integración de Servicios versión 6 en una red de arquitectura sencilla, validando la respuesta de transmisión de paquetes con un tipo de tráfico WFQ.

## **5.2.RECOMENDACIONES**

Como una recomendación principal para desarrollar o comprobar protocolos en el software de simulación NS-3 se debe simular los scripts ya disponibles en el simulador para entender el funcionamiento, así como los módulos que deben ser implementados para el desarrollo ya sea en IPV4 o IPV6.

Cuando exista algún error en el script se debe primero verificar si los módulos .h están todos incluidos como por ejemplo para activar NetAnim el modulo el #include “ns3/netanim-module.h”, para Flow-Monitor el comando es el #include “ns3/flow-monitor-module.h”

Cuando se vaya a hacer un cambio sobre un módulo ya sea .h o .cc no se debe hacer sobre los módulos creados se debe crear nuevos con nombres diferentes para evitar complicaciones y errores en el software NS-3.

## BIBLIOGRAFÍA

- Aggarwal, A., Badra, M., & Massacci, F. (2008). *New technologies, mobility and security : proceedings of NTMS '2008 conference and workshops*. IEEE.
- Ancho De Banda En El Canal Radio, M., Yasmin Becerra Sánchez, L., & Jairo Padilla Aguilar, J. (2009). *Entre Ciencia e Ingeniería, ISSN 1909-8367 Año 3. No. 5-Primer semestre de*.
- Becerra, L., Padilla, J., & Paradells, J. (5 de 10 de 2011). AN APPROACH TO RADIO CHANNEL BANDWIDTH IMPROVEMENT IN HIERARCHICAL MOBILE IP NETWORKS (HMIPv6-BI). *IEEE Latin America Transactions*, 9(5), 603-609.
- Carvajal, J., Santos, L., & Rico, D. (2013). *Movilidad en IPv6: Simulación con Network Simulator*.
- Castillo Medina, C., Felipe, /., Rodríguez, F., & Medina, C. (s.f.). *Caracterización de IPV6 Caracterización de IPv6 IPv6 Characterization*.
- Chattopadhyay, M., Jadavpur University. School of Mobile Computing and Communication, India. University Grants Commission. University of Potential Excellence Program, P., Institute of Electrical and Electronics Engineers, & IEEE Computer Society. (s.f.). *International Conference on 2015 Applications and Innovations in Mobile Computing (AIMoC) : proceedings : Jadavpur University, Kolkata, India, February 12-14, 2015*.
- De Ingeniería, E., Electrónica, E., & Redes, T. (s.f.). *ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA*.
- (s.f.). *Despliegue de IPv6 en Redes Móviles Tomás Lynch Entusiasta de IPv6*.
- Díaz, J., Demasi, M., Robles, M., & Vodopivec, G. (s.f.). *Movilidad en IPv6*.
- Duran-Faundez, C., Etter, J., & Aguilera, C. (s.f.). *Versión del Documento: 1.0.0*.
- Edición Buenos Aires, T. (2006). *Pablo Cazau INTRODUCCIÓN A LA INVESTIGACIÓN EN CIENCIAS SOCIALES*.
- El Ksimi, A., & Leghris, C. (1 de 7 de 2018). Minimising the impact of IPv6 neighbour discovery messages on energy consumption in small objects networks. *IET Networks*, 7(4).

- (s.f.). *El Método Científico y sus Etapas*, \*\*\*Ramón Ruiz\*\*\*, México 2007.
- Evaluación de IntServ6: una propuesta de calidad del servicio en redes IPv6. (2017). *Puente (Universidad Pontificia Bolivariana)*, 3(2), 7-12.
- Hernández, C., & Vicente, J. (s.f.). *Introducción al simulador de redes NS-2 Apellidos, nombre*.
- Institute of Electrical and Electronics Engineers. (2008). *2008 IEEE 10th International Symposium on Spread Spectrum Technical and Applications : proceedings : 25-28 August 2008, Bologna, Italy*. IEEE Xplore.
- IPV6 PARA OPERADORES DE RED. (s.f.).
- IPV6: The New Internet Protocol . (s.f.).
- Jaña Iturra, C. (s.f.). *Introduccion a NS2*.
- Liu, Y., Institute of Electrical and Electronics Engineers, & IEEE Circuits and Systems Society. (s.f.). *ICNC-FSKD 2017 : 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery : Guilin, Guangxi, China, 29-31 July, 2017*.
- Martínez Alayón, C., López, D., Ramírez Ochoa, J., & Gómez Tovar, R. (2 de 2013). Performance assessment of diffserv and intserv services in qos on an academic network using ns2. *TECCIENCIA*, 7(14), 65-75.
- (s.f.). *NS2 (Network Simulator version 2) Manual*.
- (2019). *ns-3 Tutorial Release ns-3.29 ns-3 project*.
- Padilla, J., Paradells, J., & Rodríguez, A. (s.f.). *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06) Supporting QoS over IPv6 wireless networks with IntServ6*.
- Padilla, J., Paradells, J., Huerta, M., & Hesselbach, X. (2005). *IntServ6: An Approach to Support QoS over IPv6 Networks*.
- Patil, D., & Patil, G. (2010). Integration of FMIPv6 in HMIPv6 to improve hand-over performance. *AIP Conference Proceedings*, 1324, págs. 166-169.
- Rodríguez-Pineda, A., & Gabriela-Maribel. (s.f.). *Área de la Energía, las Industrias y los Recursos Naturales no Renovables CARRERA DE INGENIERÍA EN SISTEMAS*.
- (s.f.). *SIMULADOR NS-2 CAROLINA JIMÉNEZ GUTIÉRREZ JORGE ALEXANDER BOTIA GONZÁLEZ UNIVERSIDAD PONTIFICIA*



*BOLIVARIANA SECCIONAL BUCARAMANGA ESCUELA DE INGENIERIAS Y ADMINISTRACION INGENIERIA ELECTRONICA BUCARAMANGA 2008.*

- Zhang, Y., & Bi, H. (2012). The simulation of hierarchical mobile IPv6 with fast handover using NS-2. *Procedia Engineering*. 37, págs. 214-217. Elsevier Ltd.

## GLOSARIO

**Ancho de banda:** Máximo Throughput posible de la red.

**BER:** Tasa de bits errados.

**HMIPv6:** Protocolo Hierarchical Mobile IPv6.

**Host:** Ordenador que funciona como punto de inicio y fin en la transferencia de datos.

**IntServ6:** Integración de Servicios version6, modelo de calidad de servicio integrados a Internet.

**IPV6:** Protocolo de Internet versión 6 que usa 128 bits.

**Jitter:** Termino que hace referencia al nivel de variación de retardo que introduce una red.

**Mbps:** Mega bits por segundo.

**Nodo:** Termino usado informalmente para referenciar a enrutadores o pc conectada a una red.

**Paquete:** Fragmento de dato pequeño y auto contenido enviado por una red.

**Pérdida de paquetes:** Porcentaje de paquetes transmitidos descartados en una red.

**Throughput:** tasa real de la información transferida.

## ANEXOS

### ANEXO 1 – DESARROLLO DEL SCRIPT EN NS3

#### 1. Inicio

2. **Módulos de inclusión.** - Hace referencia a las librerías .h, que se debe adicionar para el correcto funcionamiento del script, en el caso de una red móvil se debe implementar los siguientes:

```
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/ipv6-header.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/netanim-module.h"
#include "ns3/basic-energy-source.h"
#include "ns3/simple-device-energy-model.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/wifi-radio-energy-model.h"
#include "ns3/ipv6-static-routing-helper.h"
#include "ns3/ipv6-routing-table-entry.h"
```

**Definición de nombres.** - Define el espacio de nombres de NS3 para que permita utilizar todas las variables:

```
using namespace ns3;
```

**Registros.** - Se usa para tener un documento que contenga toda la información de cada proceso realizado.

```
NS_LOG_COMPONENT_DEFINE ("ESCENARIO01");
```

```
NS_LOG_COMPONENT_DEFINE ("HMIPV6-BI");
```

```
NS_LOG_COMPONENT_DEFINE ("INTSERV6");
```

3. **Función Principal.** – Activa los componentes de registro, así como secuencias principales del script.

```

int main (int argc, char **argv)
{
    bool verbose = false;

    CommandLine cmd;
    cmd.AddValue ("verbose", "turn on log components", verbose);
    cmd.Parse (argc, argv);

    if (verbose)
    {
        LogComponentEnable ("Ipv6ExtensionLooseRouting", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Extension", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6L3Protocol", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6StaticRouting", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Interface", LOG_LEVEL_ALL);
        LogComponentEnable ("NdiscCache", LOG_LEVEL_ALL);
    }
}

```

4. **Topologia.-** Dentro de la creación de una nueva topología se define :

**c.Create(X)**- Permite crear nodos de manera X representa el número de nodos.

```

NS_LOG_INFO ("Create nodes.");
NodeContainer c;
c.Create (13);

```

**Ptr<Node> X**- Permite crear y nombrar a los nodos.

```

NS_LOG_INFO ("Create nodes.");
Ptr<Node> CN = CreateObject<Node> ();
Ptr<Node> HA = CreateObject<Node> ();
Ptr<Node> GI = CreateObject<Node> ();
Ptr<Node> MAP = CreateObject<Node> ();
Ptr<Node> Tunel1 = CreateObject<Node> ();
Ptr<Node> Tunel2 = CreateObject<Node> ();
Ptr<Node> Tunel11 = CreateObject<Node> ();
Ptr<Node> Tunel21 = CreateObject<Node> ();
Ptr<Node> AR1 = CreateObject<Node> ();
Ptr<Node> AR2 = CreateObject<Node> ();
Ptr<Node> MN1 = CreateObject<Node> ();

```

**Uit32 t nWifi =X**.- Para la creación de nodos wifi.

```

uint32_t nWifi = 1;
CommandLine cmd;
cmd.AddValue ("nWifi", "MN", nWifi);

```

**NodeContainer**- Permite crear y gestionar nodos.

```

NS_LOG_INFO ("Create nodes.");
NodeContainer c;
c.Create (13);
NodeContainer net1 = NodeContainer (c.Get (0), c.Get (1));
NodeContainer net2 = NodeContainer (c.Get (2), c.Get (1));
NodeContainer net3 = NodeContainer (c.Get (1), c.Get (3));

```

```

NodeContainer net1 (HA, GI);
NodeContainer net2 (CN, GI);
NodeContainer net3 (GI, MAP );
NodeContainer net4 (MAP, Tunel1, Tunel2)
NodeContainer net5 (Tunel1, Tunel11);
NodeContainer net6 (Tunel2, Tunel21);
NodeContainer net7 (Tunel11, AR1);
NodeContainer net8 (Tunel21, AR2);
NodeContainer net9 (AR1, MN1);

```

**NetDeviceContainer**: Sirve para crear la red, es decir agrupar los nodos.

```

NS_LOG_INFO ("Create channels.");
CsmaHelper csma;
csma.SetDeviceAttribute ("Mtu", UintegerValue (1500));
csma.SetChannelAttribute ("DataRate", DataRateValue (5000
csma.SetChannelAttribute ("Delay", TimeValue (MilliSecond
NetDeviceContainer d1 = csma.Install (net1);
NetDeviceContainer d2 = csma.Install (net2);
NetDeviceContainer d3 = csma.Install (net3);
NetDeviceContainer d4 = csma.Install (net4);
NetDeviceContainer d5 = csma.Install (net5);
NetDeviceContainer d6 = csma.Install (net6);
NetDeviceContainer d7 = csma.Install (net7);
NetDeviceContainer d8 = csma.Install (net8);
NetDeviceContainer d9 = csma.Install (net9);
NetDeviceContainer d10 = csma.Install (net10);
NetDeviceContainer d11 = csma.Install (net11);
NetDeviceContainer d12 = csma.Install (net12);

```

**InternetStackHelper**: Sirve para instalar protocolos en los nodos.

```

NS_LOG_INFO ("Create IPv6 Internet Stack");
InternetStackHelper internetv6;
internetv6.Install (c);

```

**IPv6AddressHelper IPv6**: Sirve para dar direcciones IP a nodos.

```

NS_LOG_INFO ("Create networks and assign IPv6 Addresses.");
Ipv6AddressHelper ipv6;

ipv6.SetBase (Ipv6Address ("2001:1::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i1 = ipv6.Assign (d1);

ipv6.SetBase (Ipv6Address ("2001:2::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i2 = ipv6.Assign (d2);

ipv6.SetBase (Ipv6Address ("2001:3::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i3 = ipv6.Assign (d3);

ipv6.SetBase (Ipv6Address ("2001:4::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i4 = ipv6.Assign (d4);

ipv6.SetBase (Ipv6Address ("2001:5::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i5 = ipv6.Assign (d5);

```

**Aplicación.** - Se configura la transmisión de paquetes.

```

NS_LOG_INFO ("Create Applications.");
uint32_t packetSize = 1024;
uint32_t maxPacketCount = 1;
Time interPacketInterval = Seconds (1.0);

std::vector<Ipv6Address> routersAddress;
routersAddress.push_back (i3.GetAddress (1, 1));
routersAddress.push_back (i4.GetAddress (1, 1));
routersAddress.push_back (i5.GetAddress (1, 1));
routersAddress.push_back (i6.GetAddress (1, 1));
routersAddress.push_back (i7.GetAddress (1, 1));
routersAddress.push_back (i8.GetAddress (1, 1));
routersAddress.push_back (i9.GetAddress (1, 1));
routersAddress.push_back (i10.GetAddress (1, 1));
routersAddress.push_back (i11.GetAddress (1, 1));
routersAddress.push_back (i12.GetAddress (1, 1));
routersAddress.push_back (i2.GetAddress (0, 1));

Ping6Helper client;
/* remote address is first routers in RH0 => source routing */
client.SetRemote (i1.GetAddress (1, 1));
client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
client.SetAttribute ("Interval", TimeValue (interPacketInterval));
client.SetAttribute ("PacketSize", UintegerValue (packetSize));
client.SetRoutersAddress (routersAddress);
ApplicationContainer apps = client.Install (c.Get(12));
apps Start (Seconds (1.0));

```

5. **Simulación.** - Para realizar la simulación se debe iniciar y después de realizada destruir lo objetos creados en la simulación.

```
NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
```

6. **Análisis.** - Para analizar y determinar si el rendimiento de una red mejora se utiliza el visualizador NetAnim y el analizador Tracemetrics,

```
AsciiTraceHelper ascii;
csma.EnableAsciiAll (ascii.CreateFileStream ("escenario1.tr"));
csma.EnablePcapAll ("escenario1", true);
```

```
AnimationInterface anim ("hmipv6.xml");
anim.SetMobilityPollInterval(Seconds (1));

//NODO CN
anim.UpdateNodeDescription (c.Get(0), "CN");
anim.UpdateNodeColor (c.Get(0), 255, 0, 0);
anim.SetConstantPosition (c.Get(0), 10, 10);
anim.EnablePacketMetadata (true);
//NODO HA
anim.UpdateNodeDescription (c.Get(2), "HA");
anim.UpdateNodeColor (c.Get(2), 0, 255, 0);
anim.SetConstantPosition (c.Get(2), 10, 45);
//NODO GI
anim.UpdateNodeDescription (c.Get(1), "GI");
anim.UpdateNodeColor (c.Get(1), 0, 0, 255);
anim.SetConstantPosition (c.Get(1), 10, 30);
//NODO MAP
anim.UpdateNodeDescription (c.Get(3), "MAP");
```

7. **FIN**



## ANEXO 2 - RED CON PROTOCOLO HMIPV6

```
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/ipv6-header.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"
#include "ns3/basic-energy-source.h"
#include "ns3/simple-device-energy-model.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"
#include "ns3/wifi-radio-energy-model.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("ESCENARIO01");
int main (int argc, char **argv)
{
    bool verbose = false;
    CommandLine cmd;
    cmd.AddValue ("verbose", "turn on log components", verbose);
    cmd.Parse (argc, argv);
    if (verbose)
    {
        LogComponentEnable ("Ipv6ExtensionLooseRouting", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Extension", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6L3Protocol", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6StaticRouting", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Interface", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Interface", LOG_LEVEL_ALL);
        LogComponentEnable ("NdiscCache", LOG_LEVEL_ALL);
    }
    NS_LOG_INFO ("Create nodes.");
    NodeContainer c;
    c.Create (12);
    NodeContainer wifi;
    wifi.Create(1)
    NodeContainer net1 = NodeContainer (c.Get (0), c.Get (1));
    NodeContainer net2 = NodeContainer (c.Get (2), c.Get (1));
    NodeContainer net3 = NodeContainer (c.Get (1), c.Get (3));
    NodeContainer net4 = NodeContainer (c.Get (3), c.Get (4));
    NodeContainer net5 = NodeContainer (c.Get (3), c.Get (6));
    NodeContainer net6 = NodeContainer (c.Get (4), c.Get (5));
    NodeContainer net7 = NodeContainer (c.Get (6), c.Get (7));
    NodeContainer net8 = NodeContainer (c.Get (5), c.Get (8));
    NodeContainer net9= NodeContainer (c.Get (7), c.Get (9));
    NodeContainer net10 = NodeContainer (c.Get (8), c.Get (10));
    NodeContainer net11 = NodeContainer (c.Get (10), c.Get (11));
    NodeContainer net12 = NodeContainer (c.Get (11), wifi.Get (0));

    YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
    YansWifiPhyHelper phy = YansWifiPhyHelper::Default ();
    phy.SetChannel (channel.Create ());

    WifiHelper wifi;
    wifi.SetRemoteStationManager ("ns3::AarWifiManager");
```



```

WifiMacHelper mac;
Ssid ssid = Ssid ("ns-3-ssid");
mac.SetType ("ns3::StaWifiMac",
             "Ssid", SsidValue (ssid),
             "ActiveProbing", BooleanValue (false));
NetDeviceContainer wifiDevices;
wifiDevices = wifi.Install (phy, mac, wifiStaNodes);
mac.SetType ("ns3::ApWifiMac",
             "Ssid", SsidValue (ssid));

// Movilidad
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                              "MinX", DoubleValue (10.0),
                              "MinY", DoubleValue (10.0),
                              "DeltaX", DoubleValue (5.0),
                              "DeltaY", DoubleValue (2.0),
                              "GridWidth", UIntegerValue (5),
                              "LayoutType", StringValue ("RowFirst"));
mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                           "Bounds", RectangleValue (Rectangle (-50, 50, -25,
50)));
mobility.Install (wifi);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");

mobility.Install (wifi);
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
NS_LOG_INFO ("Create IPv6 Internet Stack");
InternetStackHelper internetv6;
internetv6.Install (c);
internetv6.Install (wifi);
Ptr<BasicEnergySource> energySource = CreateObject<BasicEnergySource>();
Ptr<WifiRadioEnergyModel> energyModel = CreateObject<WifiRadioEnergyModel>();
energySource->SetInitialEnergy (300);
energyModel->SetEnergySource (energySource);
energyModel->AppendDeviceEnergyModel (energyModel);
wifi.Get (0)->AggregateObject (energySource);
NS_LOG_INFO ("Create channels.");
CsmaHelper csma;
csma.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
csma.SetChannelAttribute ("DataRate", DataRateValue (11 000000));
csma.SetChannelAttribute ("Delay", TimeValue (Milliseconds (2)));
NetDeviceContainer d1 = csma.Install (net1);
NetDeviceContainer d2 = csma.Install (net2);
NetDeviceContainer d3 = csma.Install (net3);
NetDeviceContainer d4 = csma.Install (net4);
NetDeviceContainer d5 = csma.Install (net5);
NetDeviceContainer d6 = csma.Install (net6);
NetDeviceContainer d7 = csma.Install (net7);
NetDeviceContainer d8 = csma.Install (net8);
NetDeviceContainer d9 = csma.Install (net9);
NetDeviceContainer d10 = csma.Install (net10);
NetDeviceContainer d11 = csma.Install (net11);
NetDeviceContainer d12 = csma.Install (net12);
NS_LOG_INFO ("Create networks and assign IPv6 Addresses.");
Ipv6AddressHelper ipv6;
ipv6.SetBase (Ipv6Address ("2001:1::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i1 = ipv6.Assign (d1);
ipv6.SetBase (Ipv6Address ("2001:2::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i2 = ipv6.Assign (d2);
ipv6.SetBase (Ipv6Address ("2001:3::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i3 = ipv6.Assign (d3);

```

```

Ipv6InterfaceContainer i4 = ipv6.Assign (d4);
ipv6.SetBase (Ipv6Address ("2001:5::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i5 = ipv6.Assign (d5);
ipv6.SetBase (Ipv6Address ("2001:6::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i6 = ipv6.Assign (d6);
ipv6.SetBase (Ipv6Address ("2001:7::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i7 = ipv6.Assign (d7);
ipv6.SetBase (Ipv6Address ("2001:8::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i8 = ipv6.Assign (d8);
ipv6.SetBase (Ipv6Address ("2001:9::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i9 = ipv6.Assign (d9);
ipv6.SetBase (Ipv6Address ("2001:10::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i10 = ipv6.Assign (d10);
ipv6.SetBase (Ipv6Address ("2001:11::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i11 = ipv6.Assign (d11);
ipv6.SetBase (Ipv6Address ("2001:12::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i12 = ipv6.Assign (d12);
NS_LOG_INFO ("Create Applications.");
uint32_t packetSize = 1500;
uint32_t maxPacketCount = 1;
Time interPacketInterval = Seconds (2.0);
std::vector<Ipv6Address> routersAddress;
routersAddress.push_back (i3.GetAddress (1, 1));
routersAddress.push_back (i4.GetAddress (1, 1));
routersAddress.push_back (i5.GetAddress (1, 1));
routersAddress.push_back (i6.GetAddress (1, 1));

routersAddress.push_back (i7.GetAddress (1, 1));
routersAddress.push_back (i8.GetAddress (1, 1));
routersAddress.push_back (i9.GetAddress (1, 1));
routersAddress.push_back (i10.GetAddress (1, 1));
routersAddress.push_back (i11.GetAddress (1, 1));
routersAddress.push_back (i12.GetAddress (1, 1));
routersAddress.push_back (i2.GetAddress (0, 1));
Ping6Helper client;
/* remote address is first routers in RH0 => source routing */
client.SetRemote (i1.GetAddress (1, 1));
client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
client.SetAttribute ("Interval", TimeValue (interPacketInterval));
client.SetAttribute ("PacketSize", UintegerValue (packetSize));
client.SetRoutersAddress (routersAddress);
ApplicationContainer apps = client.Install (c.Get(0));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));
AnimationInterface anim ("escenario1.xml");
anim.SetMobilityPollInterval(Seconds (1));
Simulator::Stop (Seconds(totalTime));
Simulator::Run ();
flowMonitor->SerializeToXmlFile("escenario1flow.xml", true,true);
//NODO CN
anim.UpdateNodeDescription (c.Get(0), "CN");
anim.UpdateNodeColor (c.Get(0), 255, 0, 0);
anim.SetConstantPosition (c.Get(0), 10, 10);

//NODO HA
anim.UpdateNodeDescription (c.Get(2), "HA");
anim.UpdateNodeColor (c.Get(2), 0, 255, 0);
anim.SetConstantPosition (c.Get(2), 10, 45);
//NODO GI
anim.UpdateNodeDescription (c.Get(1), "GI");
anim.UpdateNodeColor (c.Get(1), 0, 0, 255);
anim.SetConstantPosition (c.Get(1), 10, 30);

```

```

//NODO MAP
anim.UpdateNodeDescription (c.Get(3), "MAP");
anim.UpdateNodeColor (c.Get(3), 100, 100, 0);
anim.SetConstantPosition (c.Get(3), 30, 30);
//NODO Ingreso del tunel de encapsulacion AR1
anim.UpdateNodeDescription (c.Get(4), "Tunel1");
anim.UpdateNodeColor (c.Get(4), 0, 100, 100);
anim.SetConstantPosition (c.Get(4), 15, 45);
//NODO Salida del tunel de encapsulacion AR1
anim.UpdateNodeDescription (c.Get(5), "Tunel11");
anim.UpdateNodeColor (c.Get(5), 150, 150, 100);
anim.SetConstantPosition (c.Get(5), 5, 60);
//NODO Ingreso del tunel de encapsulacion AR2
anim.UpdateNodeDescription (c.Get(6), "Tunel2");
anim.UpdateNodeColor (c.Get(6), 190, 0, 0);
anim.SetConstantPosition (c.Get(6), 25, 45);
//NODO Salida del tunel de encapsulacion AR2
anim.UpdateNodeDescription (c.Get(7), "Tunel21");
anim.UpdateNodeColor (c.Get(7), 0, 200, 0);
anim.SetConstantPosition (c.Get(7), 35, 60);
//NODO AR1
anim.UpdateNodeDescription (c.Get(8), "AR1");
anim.UpdateNodeColor (c.Get(8), 0, 0, 150);
anim.SetConstantPosition (c.Get(8), 5, 70);

//NODO AR2
anim.UpdateNodeDescription (c.Get(9), "AR2");
anim.UpdateNodeColor (c.Get(9), 175, 0, 0);
anim.SetConstantPosition (c.Get(9), 35, 70);
//NODO Ingreso del tunel de encapsulacion MN1
anim.UpdateNodeDescription (c.Get(10), "Tunel3");
anim.UpdateNodeColor (c.Get(10), 190, 0, 0);
anim.SetConstantPosition (c.Get(10), 25, 45);
//NODO Salida del tunel de encapsulacion MN1
anim.UpdateNodeDescription (c.Get(11), "Tunel31");
anim.UpdateNodeColor (c.Get(11), 0, 200, 0);
anim.SetConstantPosition (c.Get(11), 35, 60);
//NODO MN1
anim.UpdateNodeDescription (c.Get(12), "MN1");
anim.UpdateNodeColor (c.Get(12), 0, 120, 0);
anim.SetConstantPosition (c.Get(12), 5, 85);
AsciiTraceHelper ascii;
csma.EnableAsciiAll (ascii.CreateFileStream ("escenario1.tr"));
csma.EnablePcapAll ("escenario1", true);
NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
}

```

## ANEXO 3 - RED CON INTSERV6 SOBRE EL PROTOCOLO HMIPV6-BI

```
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/ipv6-header.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/internet-module.h"
#include "ns3/netanim-module.h"
#include "ns3/wifi-radio-energy-model.h"
#include "ns3/ipv6-static-routing-helper.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/ip.h"
#include "ns3/clasiffier-IntServ6.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("LooseRoutingIpv6Example");
int main (int argc, char **argv)
{
    bool verbose = false;
    CommandLine cmd;
    cmd.AddValue ("verbose", "turn on log components", verbose);
    cmd.Parse (argc, argv);
    if (verbose)
    {
        LogComponentEnable ("Ipv6ExtensionLooseRouting", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Extension", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6L3Protocol", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6StaticRouting", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Interface", LOG_LEVEL_ALL);
        LogComponentEnable ("Ipv6Interface", LOG_LEVEL_ALL);
        LogComponentEnable ("NdiscCache", LOG_LEVEL_ALL);
    }

    NS_LOG_INFO ("Create nodes.");
    Ptr<Node> CN = CreateObject<Node> ();
    Ptr<Node> HA = CreateObject<Node> ();
    Ptr<Node> GI = CreateObject<Node> ();
    Ptr<Node> MAP = CreateObject<Node> ();
    Ptr<Node> Tunel1 = CreateObject<Node> ();
    Ptr<Node> Tunel2 = CreateObject<Node> ();
    Ptr<Node> Tunel11 = CreateObject<Node> ();
    Ptr<Node> Tunel21 = CreateObject<Node> ();
    Ptr<Node> AR1 = CreateObject<Node> ();
    Ptr<Node> AR2 = CreateObject<Node> ();
    Ptr<Node> MN1 = CreateObject<Node> ();
    NodeContainer net1 (HA, GI);
    NodeContainer net2 (CN, GI);
    NodeContainer net3 (GI, MAP );
    NodeContainer net4 (MAP, Tunel1, Tunel2);
    NodeContainer net5 (Tunel1, Tunel11);
    NodeContainer net6 (Tunel2, Tunel21);
    NodeContainer net7 (Tunel11, AR1);
    NodeContainer net8 (Tunel21, AR2);
    NodeContainer net9 (AR1, MN1);
    NodeContainer all;
    all.Add (HA);
    all.Add (CN);
```



```

all.Add (GI);
all.Add (MAP);
all.Add (Tunel1);
all.Add (Tunel11);
all.Add (Tunel2);
all.Add (Tunel21);
all.Add (AR1);
all.Add (AR2);
all.Add (MN1);
NS_LOG_INFO ("Create IPv6 Internet Stack");
InternetStackHelper internetv6;
internetv6.Install (all);
NS_LOG_INFO ("Create channels.");
CsmaHelper csma;
csma.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
csma.SetChannelAttribute ("DataRate", DataRateValue (5000000));
csma.SetChannelAttribute ("Delay", TimeValue (Milliseconds (2)));
NetDeviceContainer d1 = csma.Install (net1);
NetDeviceContainer d2 = csma.Install (net2);
NetDeviceContainer d3 = csma.Install (net3);
NetDeviceContainer d4 = csma.Install (net4);
NetDeviceContainer d5 = csma.Install (net5);
NetDeviceContainer d6 = csma.Install (net6);
NetDeviceContainer d7 = csma.Install (net7);
NetDeviceContainer d8 = csma.Install (net8);
NetDeviceContainer d9 = csma.Install (net9);

NS_LOG_INFO ("Create networks and assign IPv6 Addresses.");
Ipv6AddressHelper ipv6;
ipv6.SetBase (Ipv6Address ("2001:1::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i1 = ipv6.Assign (d1);
ipv6.SetBase (Ipv6Address ("2001:2::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i2 = ipv6.Assign (d2);
ipv6.SetBase (Ipv6Address ("2001:3::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i3 = ipv6.Assign (d3);
ipv6.SetBase (Ipv6Address ("2001:4::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i4 = ipv6.Assign (d4);
ipv6.SetBase (Ipv6Address ("2001:5::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i5 = ipv6.Assign (d5);
ipv6.SetBase (Ipv6Address ("2001:6::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i6 = ipv6.Assign (d6);
ipv6.SetBase (Ipv6Address ("2001:7::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i7 = ipv6.Assign (d7);
ipv6.SetBase (Ipv6Address ("2001:8::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i8 = ipv6.Assign (d8);
ipv6.SetBase (Ipv6Address ("2001:9::"), Ipv6Prefix (64));
Ipv6InterfaceContainer i9 = ipv6.Assign (d9);
NS_LOG_INFO ("Create Applications.");
uint32_t packetSize = 1500;
uint32_t maxPacketCount = 1;
Time interPacketInterval = Seconds (1.0);
std::vector<Ipv6Address> routersAddress;
routersAddress.push_back (i3.GetAddress (1, 1));
routersAddress.push_back (i4.GetAddress (1, 1));
routersAddress.push_back (i5.GetAddress (1, 1));
routersAddress.push_back (i6.GetAddress (1, 1));
routersAddress.push_back (i7.GetAddress (1, 1));
routersAddress.push_back (i8.GetAddress (1, 1));
routersAddress.push_back (i9.GetAddress (1, 1));
routersAddress.push_back (i2.GetAddress (0, 1));
Ping6Helper client;

```

```

client.SetRemote (i1.GetAddress (1, 1));
client.SetAttribute ("MaxPackets", UIntegerValue (maxPacketCount));
client.SetAttribute ("Interval", TimeValue (interPacketInterval));
client.SetAttribute ("PacketSize", UIntegerValue (packetSize));
client.SetRoutersAddress (routersAddress);
ApplicationContainer apps = client.Install (MN1);
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));
Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();
flowMonitor->SerializeToXmlFile("escenariofinal.xml", true,true);
AnimationInterface anim ("escenariofinal.xml");
//NODO CN
anim.UpdateNodeDescription (CN, "CN");
anim.UpdateNodeColor (CN, 255, 0, 0);
anim.SetConstantPosition (CN, 10, 10);
//NODO HA
anim.UpdateNodeDescription (HA, "HA");
anim.UpdateNodeColor (HA, 0, 255, 0);
anim.SetConstantPosition (HA, 10, 45);
//NODO GI
anim.UpdateNodeDescription (GI, "GI");
anim.UpdateNodeColor (GI, 0, 0, 255);
anim.SetConstantPosition (GI, 20, 30);
//NODO MAP
anim.UpdateNodeDescription (MAP, "MAP");
anim.UpdateNodeColor (MAP, 100, 100, 0);
anim.SetConstantPosition (MAP, 30, 30);
//NODO Ingreso del tunel de encapsulacion AR1
anim.UpdateNodeDescription (Tunel1, "Tunel1");
anim.UpdateNodeColor (Tunel1, 0, 100, 100);
anim.SetConstantPosition (Tunel1, 15, 45);
//NODO Salida del tunel de encapsulacion AR1
anim.UpdateNodeDescription (Tunel11, "Tunel11");
anim.UpdateNodeColor (Tunel11, 150, 150, 100);
anim.SetConstantPosition (Tunel11, 5, 60);
//NODO Ingreso del tunel de encapsulacion AR2
anim.UpdateNodeDescription (Tunel2, "Tunel2");
anim.UpdateNodeColor (Tunel2, 190, 0, 0);
anim.SetConstantPosition (Tunel2, 25, 45);
//NODO Salida del tunel de encapsulacion AR2
anim.UpdateNodeDescription (Tunel21, "Tunel21");
anim.UpdateNodeColor (Tunel21, 0, 200, 0);
anim.SetConstantPosition (Tunel21, 35, 60);
//NODO AR1
anim.UpdateNodeDescription (AR1, "AR11");
anim.UpdateNodeColor (AR1, 0, 0, 150);
anim.SetConstantPosition (AR1, 5, 70);
//NODO AR2
anim.UpdateNodeDescription (AR2, "AR2");
anim.UpdateNodeColor (AR2, 175, 0, 0);
anim.SetConstantPosition (AR2, 35, 70);
//NODO MN1
anim.UpdateNodeDescription (MN1, "MN1");
anim.UpdateNodeColor (MN1, 0, 120, 0);
anim.SetConstantPosition (MN1, 5, 85);
AsciiTraceHelper ascii;
csma.EnableAsciiAll (ascii.CreateFileStream ("escenariofinal.tr"));
csma.EnablePcapAll ("escenariofinal", true);
NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
}

```

## Clasifier-Intserv6.h

```
#include "classifier.h"
#include "ip.h"
#include "config.h"
#include "packet.h"
#include <fstream.h>
ofstream tabla("HashIntServ6.ods");
class IntServ6Classifier : public Classifier { public:
IntServ6Classifier() : flujoint_(-1), destinoint_(-1) {
    bind("flujoint_", &flujoint_);
    bind("destinoint_", &destinoint_);
}
~IntServ6Classifier() {
};
int classify(Packet *p);
int busqueda_6(Packet *p) {
    hdr_ip* h = hdr_ip::access(p);
    return get_hash_6(h->nhash(),h->saddr(),h->daddr());
}
protected:
int set_hash_6(nsaddr_t src, int16_t sport, nsaddr_t dst, int16_t dport, int
pid) {
    int valor1;
    int indice1;
    int coll;
    coll = destinoint_; c = 0;
    indice1 = 0;

while (src<flujoint_) {
}
return 8;
}
    valor1 = hashkey_6(src,sport,dst,dport,pid);
    tabla << valor1 << "\n"; indice1 = valor1; campos6.numhash_ = valor1;
    campos6.BW_ = coll; campos6.colision_ = 0; campos6.origen_ = src;
    campos6.destino_ = dst;
    registrar_6(campos6, 0, indice1); //Llena los datos en la tabla hash
    ++src;
    ++dst;
    ++coll;
    int get_hash_6(int valorhash, nsaddr_t src, nsaddr_t dst) {
        int vfid1;
        campos6.numhash_ = valorhash; campos6.origen_ = src; campos6.destino_ = dst;
        vfid1 = registrar_6(campos6, 1, 0); return (vfid1);
    virtual int command(int argc, const char*const* argv);
    int flujoint_;
    int destinoint_;
    int keylen2_;
    int hash2_;
    int masc1_;
    int x1_;
    int x2_;
    int tablah[15];

    int idprot_;
    puntstructcampos6;
    rescol colisiones6;
    int c;
    int registrar_6(puntstruct& tab1, int a1, int indice1) {
    int a;
    int nhash1;
    int rtdocol1;
```

```

int          rtdoflow1;
int          valcol;
int          valhash;
int          y;
    a = 0;
nhash1 = 0;
rtdocol1 = 0;
rtdoflow1 = 0;
valcol = 0;
valhash = 0;
y = 0;
static int  hsttabla [2500] [5]; // [Filas] [Columnas]
//a = 0 --> Llenar tabla
//a = 1 --> Buscar Datos
//a = 2 --> Comparar datos
    if (a1 == 0) {
a = indice1;
valcol = hsttabla [a] [3]; valhash = hsttabla [a] [1]; if (valhash == indice1) {
if (valcol == 0) {
colisiones6.numhash1_ = hsttabla [a] [1]; colisiones6.origen1_ = hsttabla [a] [4];
colisiones6.destino1_ = hsttabla [a] [5]; colisiones6.BW1_ = hsttabla [a] [2];
hsttabla [a] [3] = 1;
resolver_6(colisiones6, 0); colisiones6.numhash1_ = tab1.numhash_;
colisiones6.origen1_ = tab1.origen_; colisiones6.destino1_ = tab1.destino_;
colisiones6.BW1_ = tab1.BW_; resolver_6(colisiones6, 0);
return 0;
} else if (valcol == 1) {
colisiones6.numhash1 = tab1.numhash ; colisiones6.origen1 = tab1.origen ;
colisiones6.destino1_ = tab1.destino_; colisiones6.BW1_ = tab1.BW_;
resolver_6(colisiones6, 0);
return 0;
}
} else {
return 0;
hsttabla [a] [1] = tab1.numhash_;
hsttabla [a] [2] = tab1.BW_;
hsttabla [a] [3] = tab1.colision_; hsttabla [a] [4] = tab1.origen_; hsttabla [a]
[5] = tab1.destino_; return 0;
} else if (a1 == 1) {
a = tab1.numhash_; rtdocol1 = hsttabla [a] [3]; nhash1 = hsttabla [a] [1]; if
(rtdocol1 == 0) {
rtdoflow1 = hsttabla [a][2]; return (rtdoflow1);
} else {
}
colisiones6.numhash1_ = tab1.numhash_; colisiones6.origen1_ = tab1.origen_;
colisiones6.destino1_ = tab1.destino_;
y = resolver_6(colisiones6, 1); rtdoflow1 = y;
return (rtdoflow1);
return (int) rtdoflow1;
}
return (rtdoflow1);
}
int resolver_6(rescol& punt1, int k1) { int          vcmp;
int          vcmp2;
int          maxf1;
int          vcolis1;
int          vquint1;
int          reserva1; vcmp = 0;

```



```

vcmp2 = 0;
maxf1 = 0;
vcolis1 = 0;
vquint1 = 0;
reserva1 = 0;
static int tablacoli [2500] [4]; // [Filas] [Columnas]
//k = 0 --> Llenar tabla
//k = 1 --> Buscar Datos
if (k1 == 0) {
c = c + 1;
tablacoli [c] [1] = punt1.numhash1_; tablacoli [c] [2] = punt1.origen1_;
tablacoli [c] [3] = punt1.destino1_; tablacoli [c] [4] = punt1.BW1_;
return (int) 0;
} else if (k1 == 1) {
maxf1 = flujoint_; //CAMBIAR POR EL ORIGINAL.
for (c=0;c<maxf1;c++) { // NO OLVIDAR DAR EL VALOR DE MAXF
vcmp = punt1.numhash1_; vcmp2 = punt1.origen1_; vcolis1 = tablacoli [c] [1];
if (vcmp == vcolis1) {
vquint1 = tablacoli [c] [2]; if (vcmp2 == vquint1) {
reserva1 = tablacoli [c] [4]; c = 1300;
}
}
}
return (int) reserva1;
}
return (reserva1);
}

int hashkey_6(nsaddr_t src, int16_t sport, nsaddr_t dst, int16_t dport, int
pid) {
int w;
int dir1;
int dir2;
int dir3;
dir1 = 0;
dir2 = 0;
dir3 = 0;
w = 0;
x1_ = 0;
x2_ = 0;
masc1_ = 0;
hash2_ = 0;
//Calculo hash
masc1_ = 0xfffff000; //Mask 1
x1_ = dir1;
x1_ &= masc1_;
x1_ >>= 3;
tablah[0] = x1_; //Variable 1
masc1_ = 0x0000ffff; //Mask 2 x1_ = dir1;
x1_ &= masc1_;
x1_ <<= 2;
masc1_ = 0xff000000; //Mask 3 x2_ = dir2;
x2_ &= masc1_;
x2_ >>= 6;
x1_ |= x2_;

tablah[1] = x1_; //Variable 2
masc1_ = 0x00ffff0; //Mask 4
x1_ = dir2;
x1_ &= masc1_;
x1_ >>= 1;
tablah[2] = x1_; //Variable 3

```

```

masc1_ = 0x0000000f;           //Mask 5 x1_ = dir2;
x1_ &= masc1_;
x1_ <<= 4;
masc1_ = 0xffff0000;         //Mask 6
x2_ = dir3;
x2_ &= masc1_;
x2_ >>= 4;
x1_ |= x2_;
tablah[3] = x1_; //Variable 4
masc1_ = 0x0000ffff;         //Mask 7 x1_ = dir3;
x1_ &= masc1_;
x1_ <<= 1;
masc1_ = 0xf0000000;         //Mask 8
x2_ = src;
x2_ &= masc1_;
x2_ >>= 7;
x1_ |= x2_;
tablah[4] = x1_; //Variable 5
masc1_ = 0x0fffff00;         //Mask 9
x1_ = src;
x1_ &= masc1_;
x1_ >>= 2;
tablah[5] = x1_; //Variable 6
masc1_ = 0x000000ff;         //Mask 10
x1_ = src;
x1_ &= masc1_;
x1_ <<= 3;

x2_ &= masc1_;
x2_ >>= 5;
x1_ |= x2_;
tablah[6] = x1_; //Variable 7
masc1_ = 0x000fffff;         //Mask 12
x1_ = dir1;
x1_ &= masc1_;
tablah[7] = x1_; //Variable 8
masc1_ = 0xfffff000;         //Mask 13
x1_ = dir2;
x1_ &= masc1_;
x1_ >>= 3;
tablah[8] = x1_; //Variable 9
masc1_ = 0x00000fff;         //Mask 14
x1_ = dir2;
x1_ &= masc1_;
x1_ <<= 2;
masc1_ = 0xff000000;         //Mask 15
x2_ = dir3;
x2_ &= masc1_;
x2_ >>= 6;
x1_ |= x2_;
tablah[9] = x1_; //Variable 10
masc1_ = 0x00fffff0;         //Mask 16
x1_ = dir3;
x1_ &= masc1_;
x1_ >>= 1;
tablah[10] = x1_;           //Variable 11

masc1_ = 0x0000000f;         //Mask 17
x1_ = dir3;
x1_ &= masc1_;
x1_ <<= 4;
masc1_ = 0xffff0000;         //Mask 18
x2_ = dst;
x2_ &= masc1_;

```

```

x2_ >>= 3;
x1_ |= x2_;
tablah[12] = x1_; //Variable 13
masc1_ = 0x0000ffff; //Mask 21
x1_ = sport;
x1_ &= masc1_;
x1_ <<= 2;
masc1_ = 0x0000ff00; //Mask 22
x2_ = dport;
x2_ &= masc1_;
x2_ >>= 2;
x1_ |= x2_; //Variable 14
masc1_ = 0x000000ff; //Mask 23
x1_ = dport;
x1_ &= masc1_;
x1_ <<= 2;
x2_ = pid;
x2_ &= masc1_;
x1_ |= x2_; //Variable 15
tablah[14] = x1_;
for (w=0;w<15;w++) {
hash2_ ^= tablah[w];
}
return (int) hash2_;
}
[];

```

## Ip.h

```

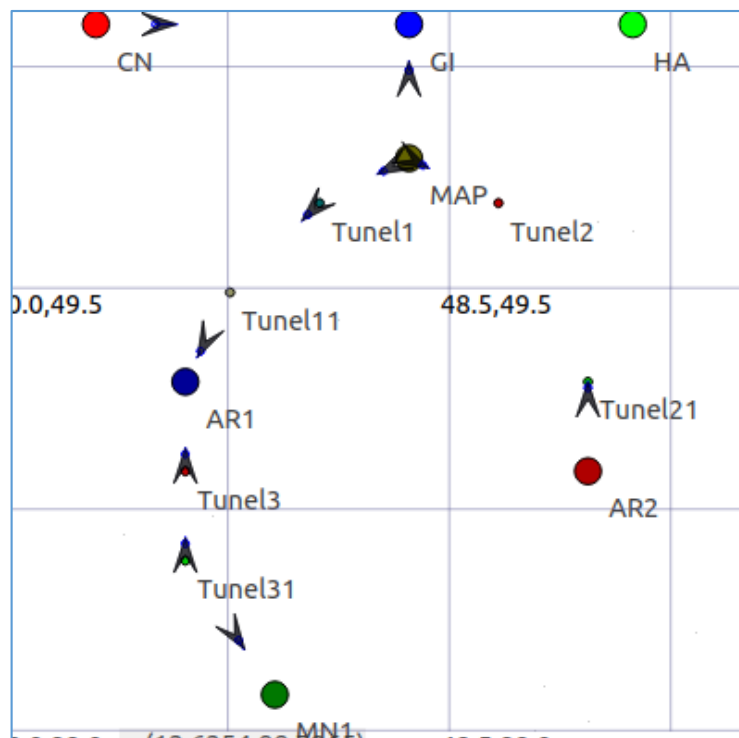
#define ns_ip_h
#include "config.h" #include "packet.h"
#define IP_HDR_LEN 20
#define IP_DEF_TTL 32
static const u_int32_t IP_BROADCAST = ((u_int32_t) 0xffffffff);
struct hdr_ip {
ns_addr_t src_;
ns_addr_t dst_;
int ttl_;
int idprot_;
int xhash_;
int fid_;
int prio_;
static int offset_;
inline static int& offset() { return offset_; } inline static hdr_ip*
access(const Packet* p) {
return (hdr_ip*) p->access(offset_);
}
ns_addr_t& src() { return (src_); } nsaddr_t& saddr() { return (src_.addr_); }
int32_t& sport() { return src_.port_;}
ns_addr_t& dst() { return (dst_); } nsaddr_t& daddr() { return (dst_.addr_); }
int32_t& dport() { return dst_.port_;} int& ttl() { return (ttl_); }
int& flowid() { return (fid_); } int& prio() { return (prio_); }
};
#endif

```

## ANEXO 4 – SIMULACIÓN DE LA RED MÓVIL CON EL PROTOCOLO HMIPV6

### Diseño de nodos de la topología del escenario1.

A continuación, se presenta el resultado de la simulación del script hmIPv6.cc el cual es el escenario1, con su respectiva visualización en el programa NetAnim se observa la ubicación de cada nodo de acuerdo a la topología implementada. Y en la se muestra la transmisión de paquetes para lograr que la red comunique a los nodos CN y MN1.

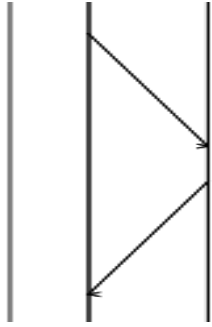
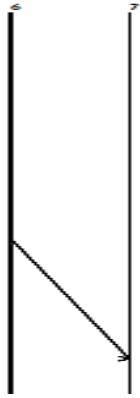
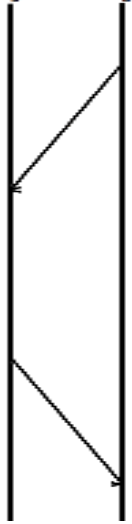


## Direccionamiento de la red del escenario1.

Nº DE NODO	DESCRIPCIÓN	DIRECCIÓN	CONEXIÓN
0	CN, es el nodo correspondiente que busca comunicarse con el nodo móvil	2001:::200:FF:FE00:1	2001:::200:FF:FE00:1 FE80::200:FF:FE00:1
2	HA, es el agente local asigna las direcciones a CN cuando va a comunicarse con MN	2001:2::200:FF:FE00:1	2001:::200:FF:FE00:3 FE80::200:FF:FE00:3
1	GI, nodo de internet del protocolo IPv6	2001:1::200:FF:FE00:1	2001:::200:FF:FE00:2 FE80::200:FF:FE00:2 2001:::200:FF:FE00:4 FE80::200:FF:FE00:4 2001:::200:FF:FE00:5 FE80::200:FF:FE00:5
3	MAP, es el encargado de gestionar la movilidad de MN encargado de encapsular y desencapsular los paquetes por lo que los re direcciona a los nodos punto de entrada de túnel(TUNEL1 , TUNEL2)	2001:3::200:FF:FE00:1	2001:::200:FF:FE00:6 FE80::200:FF:FE00:6 2001:::200:FF:FE00:7 FE80::200:FF:FE00:7 2001:::200:FF:FE00:9 FE80::200:FF:FE00:9
4	TUNEL1, nodo entrada de túnel al que llegan el paquete original desde el MAP para AR1	2001:4::200:FF:FE00:1	2001:::200:FF:FE00:8 FE80::200:FF:FE00:B 2001:::200:FF:FE00:8 FE80::200:FF:FE00:B
6	TUNEL2, nodo entrada de túnel al que llegan el paquete original desde el MAP para AR2	2001:6::200:FF:FE00:1	2001:::200:FF:FE00:A FE80::200:FF:FE00:A 2001:::200:FF:FE00:D FE80::200:FF:FE00:D
5	TUNEL11, nodo salida de túnel al que llegan el paquete encapsulado de TUNEL1, para AR1.	2001:5::200:FF:FE00:1	2001:::200:FF:FE00:C FE80::200:FF:FE00:C 2001:::200:FF:FE00:F FE80::200:FF:FE00:F
7	TUNEL21, nodo salida de túnel al que llegan el paquete encapsulado de TUNEL2 , para AR2.	2001:7::200:FF:FE00:1	2001:::200:FF:FE00:E FE80::200:FF:FE00:E 2001:::200:FF:FE00:11 FE80::200:FF:FE00:11
8	AR1 router acceso 1 donde llegan los paquetes tunelizados desde TUNEL11	2001:8::200:FF:FE00:1	2001:::200:FF:FE00:10 FE80::200:FF:FE00:10 2001:::200:FF:FE00:13 FE80::200:FF:FE00:13
9	AR2 router acceso 2 donde llegan los paquetes tunelizados desde TUNEL21	2001:9::200:FF:FE00:1	2001:::200:FF:FE00:12 FE80::200:FF:FE00:12
10	TUNEL3 Encapsula el paquete que viene desde el AR1	2001:10::200:FF:FE00:1	2001:::200:FF:FE00:14 FE80::200:FF:FE00:14 2001:::200:FF:FE00:15 FE80::200:FF:FE00:15
11	TUNEL31 nodo salida de túnel para comunicarse con MN1	2001:11::200:FF:FE00:1	2001:::200:FF:FE00:16 FE80::200:FF:FE00:16 2001:::200:FF:FE00:17 FE80::200:FF:FE00:17
12	MN1 nodo móvil con el que CN se comunica.	2001:12::200:FF:FE00:1	2001:::200:FF:FE00:18 FE80::200:FF:FE00:18

**Tiempos de comunicación entre nodos de la red del escenario1 en el visualizador NetAnim**

TRANSMISIÓN	PAQUETE	TIEMPOS																					
CN-HA	<pre> sequenceDiagram     participant N1 as 1     participant N2 as 2     N1-&gt;&gt;N2:      </pre>	<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>0.003343</td> </tr> <tr> <td>1</td> <td>2</td> <td>0.003343</td> </tr> </tbody> </table>	From Id	To Id	Tx	1	2	0.003343	1	2	0.003343												
From Id	To Id	Tx																					
1	2	0.003343																					
1	2	0.003343																					
CN-GI	<pre> sequenceDiagram     participant N0 as 0     participant N1 as 1     N0-&gt;&gt;N1:      N0-&gt;&gt;N1:      </pre>	<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>3 1</td> <td>0</td> <td>0.00415</td> </tr> <tr> <td>4 1</td> <td>0</td> <td>0.00415</td> </tr> </tbody> </table>	From Id	To Id	Tx	3 1	0	0.00415	4 1	0	0.00415												
From Id	To Id	Tx																					
3 1	0	0.00415																					
4 1	0	0.00415																					
GI-MAP	<pre> sequenceDiagram     participant N1 as 1     participant N2 as 2     N1-&gt;&gt;N2:      N1-&gt;&gt;N2:      N2-&gt;&gt;N1:      N2-&gt;&gt;N1:      </pre>	<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>3</td> <td>0.001</td> </tr> <tr> <td>1</td> <td>3</td> <td>0.001</td> </tr> <tr> <td>3</td> <td>1</td> <td>0.006</td> </tr> <tr> <td>3</td> <td>1</td> <td>0.006</td> </tr> </tbody> </table>	From Id	To Id	Tx	1	3	0.001	1	3	0.001	3	1	0.006	3	1	0.006						
From Id	To Id	Tx																					
1	3	0.001																					
1	3	0.001																					
3	1	0.006																					
3	1	0.006																					
MAP-TUNEL1 TUNEL2	<pre> sequenceDiagram     participant N3 as 3     participant N4 as 4     participant N6 as 6     N4-&gt;&gt;N3:      N4-&gt;&gt;N3:      N6-&gt;&gt;N3:      N6-&gt;&gt;N3:      N3-&gt;&gt;N4:      N3-&gt;&gt;N4:      </pre>	<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>3</td> <td>0.002</td> </tr> <tr> <td>4</td> <td>3</td> <td>0.002</td> </tr> <tr> <td>6</td> <td>3</td> <td>0.004379</td> </tr> <tr> <td>6</td> <td>3</td> <td>0.004379</td> </tr> <tr> <td>3</td> <td>4</td> <td>0.006797</td> </tr> <tr> <td>3</td> <td>4</td> <td>0.006797</td> </tr> </tbody> </table>	From Id	To Id	Tx	4	3	0.002	4	3	0.002	6	3	0.004379	6	3	0.004379	3	4	0.006797	3	4	0.006797
From Id	To Id	Tx																					
4	3	0.002																					
4	3	0.002																					
6	3	0.004379																					
6	3	0.004379																					
3	4	0.006797																					
3	4	0.006797																					

TRANSMISIÓN	PAQUETE	TIEMPOS															
TUNEL1-TUNEL11		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>5</td> <td>0.002</td> </tr> <tr> <td>4</td> <td>5</td> <td>0.002</td> </tr> <tr> <td>5</td> <td>4</td> <td>0.004822</td> </tr> <tr> <td>5</td> <td>4</td> <td>0.004822</td> </tr> </tbody> </table>	From Id	To Id	Tx	4	5	0.002	4	5	0.002	5	4	0.004822	5	4	0.004822
From Id	To Id	Tx															
4	5	0.002															
4	5	0.002															
5	4	0.004822															
5	4	0.004822															
TUNEL2-TUNEL21		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>7</td> <td>0.004</td> </tr> <tr> <td>6</td> <td>7</td> <td>0.004</td> </tr> </tbody> </table>	From Id	To Id	Tx	6	7	0.004	6	7	0.004						
From Id	To Id	Tx															
6	7	0.004															
6	7	0.004															
TUNEL11-AR1		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>5</td> <td>0.001</td> </tr> <tr> <td>8</td> <td>5</td> <td>0.001</td> </tr> <tr> <td>5</td> <td>8</td> <td>0.006</td> </tr> <tr> <td>5</td> <td>8</td> <td>0.006</td> </tr> </tbody> </table>	From Id	To Id	Tx	8	5	0.001	8	5	0.001	5	8	0.006	5	8	0.006
From Id	To Id	Tx															
8	5	0.001															
8	5	0.001															
5	8	0.006															
5	8	0.006															

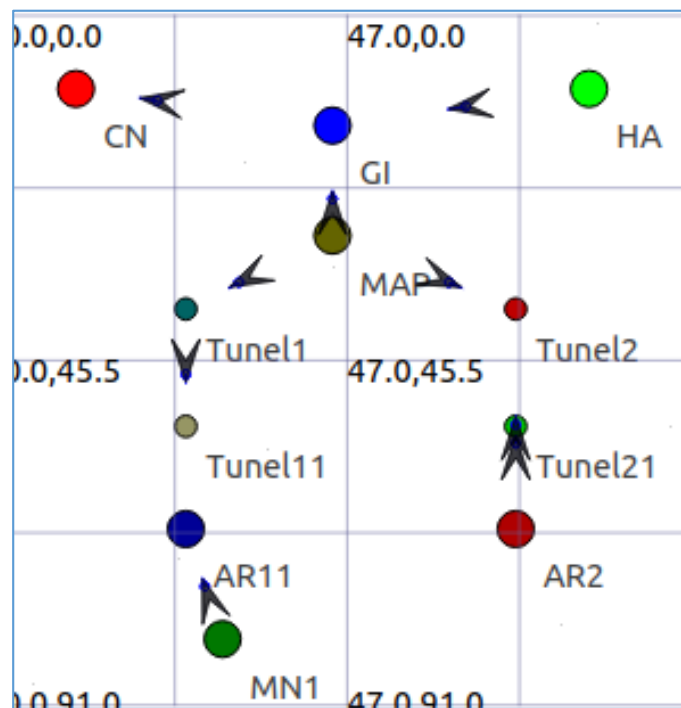
TRANSMISIÓN	PAQUETE	TIEMPO																					
TUNEL21-AR2		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>9</td> <td>0.001</td> </tr> <tr> <td>7</td> <td>9</td> <td>0.001</td> </tr> <tr> <td>9</td> <td>7</td> <td>0.0054102</td> </tr> <tr> <td>9</td> <td>7</td> <td>0.0054102</td> </tr> </tbody> </table>	From Id	To Id	Tx	7	9	0.001	7	9	0.001	9	7	0.0054102	9	7	0.0054102						
From Id	To Id	Tx																					
7	9	0.001																					
7	9	0.001																					
9	7	0.0054102																					
9	7	0.0054102																					
AR1-TUNEL3-TUNEL31		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>10</td> <td>0</td> </tr> <tr> <td>8</td> <td>10</td> <td>0</td> </tr> <tr> <td>10</td> <td>11</td> <td>0.001</td> </tr> <tr> <td>10</td> <td>11</td> <td>0.001</td> </tr> <tr> <td>11</td> <td>10</td> <td>0.004</td> </tr> <tr> <td>11</td> <td>10</td> <td>0.004</td> </tr> </tbody> </table>	From Id	To Id	Tx	8	10	0	8	10	0	10	11	0.001	10	11	0.001	11	10	0.004	11	10	0.004
From Id	To Id	Tx																					
8	10	0																					
8	10	0																					
10	11	0.001																					
10	11	0.001																					
11	10	0.004																					
11	10	0.004																					
TUNEL31-MN1		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>11</td> <td>0.0039812</td> </tr> <tr> <td>12</td> <td>11</td> <td>0.0039812</td> </tr> <tr> <td>11</td> <td>12</td> <td>0.006134</td> </tr> <tr> <td>11</td> <td>12</td> <td>0.006134</td> </tr> </tbody> </table>	From Id	To Id	Tx	12	11	0.0039812	12	11	0.0039812	11	12	0.006134	11	12	0.006134						
From Id	To Id	Tx																					
12	11	0.0039812																					
12	11	0.0039812																					
11	12	0.006134																					
11	12	0.006134																					



## ANEXO 5 – SIMULACIÓN DE LA RED MÓVIL CON INTSERV6 SOBRE PROTOCOLO HMIPV6-BI

### Diseño de nodos de la topología de HMIPV6-BI escenario final



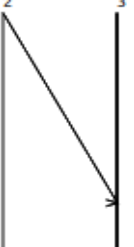
En la figura, se muestra la ubicación de los nodos según el programa NetAnim del script HMIPv6-BI.cc de acuerdo a la topología para verificar la transmisión de paquetes entre CN y MN1.

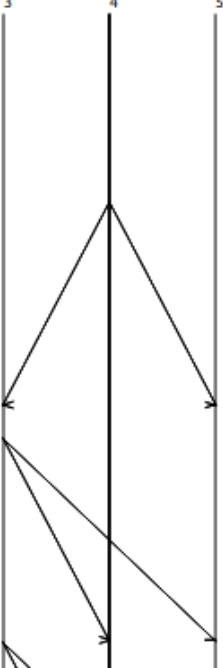




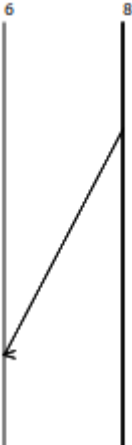
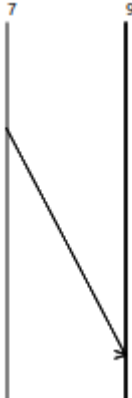
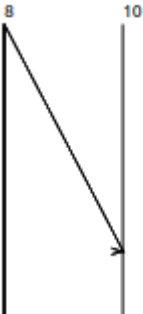
## Direccionamiento de la red del escenario final.

Nº DE NODO	DESCRIPCIÓN	DIRECCIÓN	CONEXIÓN
0	CN, es el nodo correspondiente que busca comunicarse con el nodo móvil	2001:::200:FF:FE00:1	2001:::200:FF:FE00:3 FE80::200:FF:FE00:3
1	HA, es el agente local asigna las direcciones a CN cuando va a comunicarse con MN	2001:1::200:FF:FE00:1	2001:::200:FF:FE00:1 FE80::200:FF:FE00:1
2	GI, nodo de internet del protocolo IPv6	2001:2::200:FF:FE00:1	2001:::200:FF:FE00:2 FE80::200:FF:FE00:2 2001:::200:FF:FE00:4 FE80::200:FF:FE00:4 2001:::200:FF:FE00:5 FE80::200:FF:FE00:5
3	MAP, es el encargado de gestionar la movilidad de MN encargado de encapsular y desencapsular los paquetes por lo que los re direcciona a los nodos punto de entrada de túnel(TUNEL1 , TUNEL2)	2001:3::200:FF:FE00:1	2001:::200:FF:FE00:6 FE80::200:FF:FE00:6 2001:::200:FF:FE00:7 FE80::200:FF:FE00:7
4	TUNEL1, nodo entrada de túnel al que llegan el paquete original desde el MAP para AR1	2001:4::200:FF:FE00:1	2001:::200:FF:FE00:8 FE80::200:FF:FE00:B 2001:::200:FF:FE00:8 FE80::200:FF:FE00:B
5	TUNEL2, nodo entrada de túnel al que llegan el paquete original desde el MAP para AR2	2001:5::200:FF:FE00:1	2001:::200:FF:FE00:9 FE80::200:FF:FE00:9 2001:::200:FF:FE00:C FE80::200:FF:FE00:C
6	TUNEL11, nodo salida de túnel al que llegan el paquete encapsulado de TUNEL1, para AR1.	2001:6::200:FF:FE00:1	2001:::200:FF:FE00:B FE80::200:FF:FE00: B 2001:::200:FF:FE00:E FE80::200:FF:FE00:E
7	TUNEL21, nodo salida de túnel al que llegan el paquete encapsulado de TUNEL2 , para AR2.	2001:7::200:FF:FE00:1	2001:::200:FF:FE00:D FE80::200:FF:FE00: D 2001:::200:FF:FE00:10 FE80::200:FF:FE00:10
8	AR1 router acceso 1 donde llegan los paquetes tunelizados desde TUNEL11	2001:8::200:FF:FE00:1	2001:::200:FF:FE00:12 FE80::200:FF:FE00:12 2001:::200:FF:FE00:F FE80::200:FF:FE00:F
9	AR2 router acceso 2 donde llegan los paquetes tunelizados desde TUNEL21 hacia el MN1	2001:9::200:FF:FE00:1	2001:::200:FF:FE00:11 FE80::200:FF:FE00:11
10	MN1 nodo móvil con el que CN se comunica.	2001:10::200:FF:FE00:1	2001:::200:FF:FE00:13 FE80::200:FF:FE00:13

**Tiempos de comunicación entre nodos de la red del escenario con HMIPV6-BI en el visualizador NetAnim**

TRANSMISIÓN	PAQUETE	TIEMPOS																					
HA-CN		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>0.001</td> </tr> <tr> <td>1</td> <td>2</td> <td>0.001</td> </tr> <tr> <td>1</td> <td>2</td> <td>0.003343</td> </tr> <tr> <td>1</td> <td>2</td> <td>0.003343</td> </tr> </tbody> </table>	From Id	To Id	Tx	1	2	0.001	1	2	0.001	1	2	0.003343	1	2	0.003343						
From Id	To Id	Tx																					
1	2	0.001																					
1	2	0.001																					
1	2	0.003343																					
1	2	0.003343																					
CN-GI		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> <td>0.002</td> </tr> <tr> <td>0</td> <td>2</td> <td>0.002</td> </tr> <tr> <td>2</td> <td>0</td> <td>0.00428</td> </tr> <tr> <td>2</td> <td>0</td> <td>0.00428</td> </tr> <tr> <td>2</td> <td>0</td> <td>0.006567</td> </tr> <tr> <td>2</td> <td>0</td> <td>0.006567</td> </tr> </tbody> </table>	From Id	To Id	Tx	0	2	0.002	0	2	0.002	2	0	0.00428	2	0	0.00428	2	0	0.006567	2	0	0.006567
From Id	To Id	Tx																					
0	2	0.002																					
0	2	0.002																					
2	0	0.00428																					
2	0	0.00428																					
2	0	0.006567																					
2	0	0.006567																					
GI-MAP		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>3</td> <td>0</td> </tr> <tr> <td>2</td> <td>3</td> <td>0</td> </tr> <tr> <td>2</td> <td>3</td> <td>0.005</td> </tr> <tr> <td>2</td> <td>3</td> <td>0.005</td> </tr> <tr> <td>3</td> <td>2</td> <td>0.007314</td> </tr> <tr> <td>3</td> <td>2</td> <td>0.007314</td> </tr> </tbody> </table>	From Id	To Id	Tx	2	3	0	2	3	0	2	3	0.005	2	3	0.005	3	2	0.007314	3	2	0.007314
From Id	To Id	Tx																					
2	3	0																					
2	3	0																					
2	3	0.005																					
2	3	0.005																					
3	2	0.007314																					
3	2	0.007314																					

TRANSMISIÓN	PAQUETES	TIEMPOS																											
MAP-TUNEL1-TUNEL2		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr><td>4</td><td>3</td><td>0.002</td></tr> <tr><td>4</td><td>3</td><td>0.002</td></tr> <tr><td>4</td><td>5</td><td>0.002</td></tr> <tr><td>4</td><td>5</td><td>0.002</td></tr> <tr><td>3</td><td>4</td><td>0.004498</td></tr> <tr><td>3</td><td>4</td><td>0.004498</td></tr> <tr><td>3</td><td>5</td><td>0.004498</td></tr> <tr><td>3</td><td>5</td><td>0.004498</td></tr> </tbody> </table>	From Id	To Id	Tx	4	3	0.002	4	3	0.002	4	5	0.002	4	5	0.002	3	4	0.004498	3	4	0.004498	3	5	0.004498	3	5	0.004498
From Id	To Id	Tx																											
4	3	0.002																											
4	3	0.002																											
4	5	0.002																											
4	5	0.002																											
3	4	0.004498																											
3	4	0.004498																											
3	5	0.004498																											
3	5	0.004498																											
TUNEL1-TUNEL11		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr><td>4</td><td>6</td><td>0.002</td></tr> <tr><td>4</td><td>6</td><td>0.002</td></tr> <tr><td>6</td><td>4</td><td>0.004454</td></tr> <tr><td>6</td><td>4</td><td>0.004454</td></tr> </tbody> </table>	From Id	To Id	Tx	4	6	0.002	4	6	0.002	6	4	0.004454	6	4	0.004454												
From Id	To Id	Tx																											
4	6	0.002																											
4	6	0.002																											
6	4	0.004454																											
6	4	0.004454																											
TUNEL2-TUNEL21		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr><td>5</td><td>7</td><td>0.004</td></tr> <tr><td>5</td><td>7</td><td>0.004</td></tr> <tr><td>7</td><td>5</td><td>0.008</td></tr> <tr><td>7</td><td>5</td><td>0.008</td></tr> </tbody> </table>	From Id	To Id	Tx	5	7	0.004	5	7	0.004	7	5	0.008	7	5	0.008												
From Id	To Id	Tx																											
5	7	0.004																											
5	7	0.004																											
7	5	0.008																											
7	5	0.008																											

TRANSMISIÓN	PAQUETES	TIEMPOS																					
<p>TUNEL11-AR1</p>		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>6</td> <td>0.001</td> </tr> <tr> <td>8</td> <td>6</td> <td>0.001</td> </tr> <tr> <td>6</td> <td>8</td> <td>0.006</td> </tr> <tr> <td>6</td> <td>8</td> <td>0.006</td> </tr> </tbody> </table>	From Id	To Id	Tx	8	6	0.001	8	6	0.001	6	8	0.006	6	8	0.006						
From Id	To Id	Tx																					
8	6	0.001																					
8	6	0.001																					
6	8	0.006																					
6	8	0.006																					
<p>TUNEL21-AR2</p>		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>9</td> <td>0.001</td> </tr> <tr> <td>7</td> <td>9</td> <td>0.001</td> </tr> <tr> <td>9</td> <td>7</td> <td>0.003967</td> </tr> <tr> <td>9</td> <td>7</td> <td>0.003967</td> </tr> <tr> <td>9</td> <td>7</td> <td>0.0062382</td> </tr> <tr> <td>9</td> <td>7</td> <td>0.0062382</td> </tr> </tbody> </table>	From Id	To Id	Tx	7	9	0.001	7	9	0.001	9	7	0.003967	9	7	0.003967	9	7	0.0062382	9	7	0.0062382
From Id	To Id	Tx																					
7	9	0.001																					
7	9	0.001																					
9	7	0.003967																					
9	7	0.003967																					
9	7	0.0062382																					
9	7	0.0062382																					
<p>AR1-MN1</p>		<table border="1"> <thead> <tr> <th>From Id</th> <th>To Id</th> <th>Tx</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>10</td> <td>0</td> </tr> <tr> <td>8</td> <td>10</td> <td>0</td> </tr> <tr> <td>10</td> <td>8</td> <td>0.007</td> </tr> <tr> <td>10</td> <td>8</td> <td>0.007</td> </tr> </tbody> </table>	From Id	To Id	Tx	8	10	0	8	10	0	10	8	0.007	10	8	0.007						
From Id	To Id	Tx																					
8	10	0																					
8	10	0																					
10	8	0.007																					
10	8	0.007																					