



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE TELECOMUNICACIONES**

**INFORME FINAL DEL PROYECTO DE INTEGRACIÓN  
CURRICULAR, PROYECTO DE INVESTIGACIÓN**

**TEMA:**

**“APLICACIÓN Y EVALUACIÓN DE CALIDAD DE SERVICIO (QoS)  
SOBRE UNA RED INALÁMBRICA DEFINIDA POR SOFTWARE  
(SDWN)”**

**Trabajo de grado previo a la obtención del título de Ingeniera en Telecomunicaciones**

**AUTOR:**

Karla Nayeli Moncayo Chávez

**DIRECTOR:**

Msc. Hernán Mauricio Domínguez Limaico

**Ibarra, febrero 2025**

**UNIVERSIDAD TECNICA DEL NORTE BIBLIOTECA  
UNIVERSITARIA**

**IDENTIFICACIÓN DE LA OBRA**

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

<b>DATOS DEL CONTACTO</b>	
<b>CÉDULA DE IDENTIDAD</b>	0450095989
<b>APELLIDOS Y NOMBRES</b>	Moncayo Chávez Karla Nayeli
<b>DIRECCIÓN</b>	Ibarra, Av. 17 de Julio frente a Polideportivo Recreacional
<b>E-MAIL</b>	knmoncayoc@utn.edu.ec   karla15moncayo@gmail.com
<b>TELÉFONO FIJO</b>	<b>TELÉFONO MÓVIL</b> 0959492107

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO</b>	“Aplicación y Evaluación de Calidad de Servicio (QoS) sobre una Red Inalámbrica Definida por Software (SDWN)”.
<b>AUTOR</b>	Moncayo Chávez Karla Nayeli
<b>FECHA</b>	15/02/2025
<b>PROGRAMA</b>	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
<b>TÍTULO</b>	Ingeniera en Telecomunicaciones
<b>DIRECTOR</b>	Msc. Hernán Mauricio Domínguez Limaico

**AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, Karla Nayeli Moncayo Chávez, con cédula de identidad Nro. 0450095989, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 15 días del mes de febrero de 2025

**EL AUTOR:**

.....*Karla Moncayo*.....

Moncayo Chávez Karla Nayeli

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 15 días del mes de febrero de 2025

### EL AUTOR:

.....*Karla Moncayo*.....

Moncayo Chávez Karla Nayeli



**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE  
INTEGRACIÓN CURRICULAR**

Ibarra, 14 de febrero del 2025

Magíster Hernán Mauricio Domínguez Limaico  
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:


Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.




*MSc. Hernán Mauricio Domínguez Limaico*  
*C.C.: 1002379301*

### APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificador del trabajo de Integración Curricular “Aplicación y Evaluación de Calidad de servicio (QoS) sobre una Red Inalámbrica Definida por Software (SDWN)” elaborado por Moncayo Chávez Karla Nayeli, previo a la obtención del título de Ingeniera en Telecomunicaciones, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

(f):   
MSc. Hernán Mauricio Domínguez Limaico  
C.C.: 1002379301

(f):   
MSc. Carlos Alberto Vásquez Ayala  
C.C.: 1002424982

**DEDICATORIA**

*Con todo mi cariño, dedico este trabajo a mi familia, quienes han sido mi fuente de inspiración, motivación y confianza. Sin duda, este logro es el reflejo del amor, las palabras de aliento y la sabiduría que todos ustedes han sembrado en mí. ¡Los amo con mi alma!*

*Karla Nayeli Moncayo Chávez*

## AGRADECIMIENTO

*A mis padres Carlos y Mariana, porque cada travesía que he culminado ha sido gracias a su esfuerzo y apoyo incondicional. ¡Todo lo maravilloso de mi vida se lo debo a ustedes!*

*A mis tías Bertha y Lorgia, quienes desde mi niñez me han impulsado a salir adelante. Agradezco su influencia en mi vida y estoy segura de que sus oraciones iluminaron mi camino hacia este acontecimiento.*

*A mis hermanos Lenin, Bryan, Damián, Mathyas y Giomara. Su compañía invaluable y sus enseñanzas han sido esenciales durante todo este recorrido.*

*A mis amigos Mafer, Jessy, Karen, Ariel, Keneth, Ismael, Edwin, Marco R. y Marco L. Su amistad y compañerismo hicieron que esta experiencia sea inolvidable. ¡Gracias por cada momento compartido!*

*A los prestigiosos docentes que conforman la carrera de Telecomunicaciones en la UTN, en especial a Ing. Mauricio Domínguez e Ing. Carlos Vásquez, por ser parte fundamental en desarrollo de este trabajo. Gracias por su tiempo, paciencia y conocimientos compartidos.*

## ÍNDICE DE CONTENIDO

<b>LISTA DE SIGLAS.....</b>	<b>1</b>
<b>RESUMEN EJECUTIVO .....</b>	<b>3</b>
<b>ABSTRACT.....</b>	<b>4</b>
<b>1. CAPÍTULO I: Antecedentes.....</b>	<b>5</b>
1.1. Tema.....	5
1.2. Problema .....	5
1.3. Objetivos.....	8
1.3.1. Objetivo general .....	8
1.3.2. Objetivos específicos.....	8
1.4. Alcance .....	8
1.5. Justificación .....	11
<b>2. CAPÍTULO II: Estado del arte.....</b>	<b>13</b>
2.1. Estándar IEEE 802.11n.....	13
2.1.1. Capa Control de Acceso al Medio (MAC).....	14
2.1.1.1. Agregación de tramas. ....	15
2.1.2. Capa Física (PHY).....	17
2.1.2.1. MIMO.....	18
2.2. Calidad de servicio (QoS).....	18
2.2.1. Modelos de QoS.....	22
2.2.1.1. Best-Effort (BE).....	23
2.2.1.2. Modelo de Servicios Integrados (IntServ). ....	24
2.2.1.3. Modelo de Servicios Diferenciados (DiffServ). ....	25
2.2.2. Servicios y sus protocolos.....	28
2.2.3. Normativas de QoS .....	29
2.2.3.1. ITU-T E.800.....	30
2.2.3.2. ITU-T Y.1540.....	31
2.2.3.3. ITU-T Y.1541.....	37
2.3. Redes inalámbricas definidas por software (SDWN).....	38
2.3.1. Arquitectura SDWN.....	40
2.3.2. Componentes de red SDWN .....	41
2.3.2.1. Controlador SDN.....	41
2.3.2.2. Interfaces SDN.....	42
2.3.2.3. Puntos de acceso.....	43
2.3.3. Gestión de calidad de servicio en SDWN.....	43

<b>3. CAPÍTULO III: Diseño de red e implementación de QoS .....</b>	<b>45</b>
3.1. Fases de desarrollo .....	45
3.2. Establecimiento de prototipo SDWN.....	45
3.2.1. Topología lógica de red SDWN.....	47
3.2.2. Instalación de controlador Ryu .....	48
3.2.3. Configuración de APs programables .....	52
3.2.3.1. Instalación de firmware OpenWrt en RPi 3B .....	52
3.2.3.2. Instalación de módulo Open Virtual Switch en el sistema OpenWrt.....	56
3.2.3.3. Creación de puente con Open Virtual Switch .....	58
3.2.3.4. Establecimiento de comunicación entre el controlador y los APs .....	61
3.2.3.5. Configuración de red inalámbrica en APs .....	64
3.3. Planteamiento y Aplicación de Políticas de QoS.....	69
3.3.1. Planteamiento de QoS en la red SDWN .....	69
3.3.2. Aplicación de QoS en la red .....	77
3.3.2.1. Establecimiento de conexión a OVSDB .....	77
3.3.2.2. Configuración de colas para cada AP .....	81
3.3.2.3. Parametrización de marcaje en la red .....	84
3.3.2.4. Clasificación de tráfico .....	86
3.3.2.5. Visualización de marcaje con Wireshark .....	88
<b>4. CAPÍTULO IV: Pruebas y análisis de funcionamiento .....</b>	<b>92</b>
4.1. Planteamiento de pruebas conforme a ITU-T Y.1540.....	92
4.1.1. Topología de pruebas .....	93
4.1.2. Parámetros de rendimiento en transferencia de paquetes IP.....	94
4.1.3. Configuración de pruebas para evaluación de QoS .....	95
4.1.3.1. Escenarios evaluados antes de aplicar QoS .....	96
4.1.3.1.1. Escenario 1: Tráfico ligero.....	96
4.1.3.1.2. Escenario 2: Tráfico moderado .....	104
4.1.3.1.3. Escenario 3: Tráfico intenso.....	110
4.1.3.1.4. Escenario 4: Sobrecarga de red.....	116
4.1.3.2. Escenarios evaluados después de aplicar QoS.....	122
4.1.3.2.1. Escenario 1: Tráfico ligero.....	122
4.1.3.2.2. Escenario 2: Tráfico moderado .....	128
4.1.3.2.3. Escenario 3: Tráfico intenso.....	133
4.1.3.2.4. Escenario 4: Sobrecarga de red.....	139
4.1.4. Resumen y comparación de pruebas de rendimiento.....	144

<b>5. CONCLUSIONES .....</b>	<b>153</b>
<b>6. RECOMENDACIONES .....</b>	<b>154</b>
<b>7. REFERENCIAS.....</b>	<b>156</b>
<b>8. ANEXOS.....</b>	<b>162</b>
8.1. Anexo A: Instalación y configuración de servicios multimedia .....	162
8.1.1. Servicio VOD con Jellyfin.....	162
8.1.2. Servicio de Transferencia de Archivos (FTP).....	168
8.1.3. Servicio de Voz sobre IP (VoIP) con Issabel.....	172

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Topología de red física SDWN.....	10
<b>Figura 2</b> Transmisión sin agregación de tramas. ....	15
<b>Figura 3</b> Agregación de tramas A-MSDU.....	16
<b>Figura 4</b> Agregación de tramas A-MPDU.....	17
<b>Figura 5</b> Agregación A-MSDU y A-MPDU combinadas. ....	17
<b>Figura 6</b> Sistema MIMO de 4x4. ....	18
<b>Figura 7</b> Diferencia entre Latencia y RTT. ....	22
<b>Figura 8</b> Ejemplo de Jitter. ....	22
<b>Figura 9</b> Diferencias entre las cabeceras IPv4 e IPv6.....	24
<b>Figura 10</b> Esquema de funcionamiento de IntServ. ....	25
<b>Figura 11</b> Estructura de campo ToS/TC en paquetes IP.....	27
<b>Figura 12</b> Los cuatro polos de la Calidad de Servicio. ....	30
<b>Figura 13</b> Resultados posibles en la transferencia de paquetes IP. ....	33
<b>Figura 14</b> Alcance de la recomendación ITU-T Y.1540. ....	37
<b>Figura 15</b> Arquitectura SDN. ....	41
<b>Figura 16</b> Metodología para desarrollo de proyecto.....	45
<b>Figura 17</b> Topología lógica de red SDWN. ....	47
<b>Figura 18</b> Fallo en la inicialización del controlador Ryu.....	48
<b>Figura 19</b> Dependencias de Python3 necesarias para controlador Ryu. ....	50
<b>Figura 20</b> Verificación de paquetes instalados.....	50
<b>Figura 21</b> Instalación de Ryu con pip3. ....	50
<b>Figura 22</b> Clonación de ejemplos de Python para controlador Ryu. ....	51
<b>Figura 23</b> Inicialización de controlador Ryu. ....	51
<b>Figura 24</b> Diagrama de conexión de OpenWrt y OVS en RPi 3B.....	52
<b>Figura 25</b> Descarga de firmware OpenWrt para Raspberry Pi3B.....	52
<b>Figura 26</b> Uso de adaptador para lectura de tarjeta microSD en PC. ....	53
<b>Figura 27</b> Escritura de Firmware con Raspberry Pi Imager.....	54
<b>Figura 28</b> Configuración de sistema OpenWrt mediante acceso remoto.....	54
<b>Figura 29</b> Cambio de contraseña de acceso para administración de sistema OpenWrt.....	55
<b>Figura 30</b> Acceso a sistema OpenWrt desde navegador. ....	55
<b>Figura 31</b> Configuración de interfaz LAN en modo cliente DHCP.....	56
<b>Figura 32</b> Actualización del sistema OpenWrt. ....	56

<b>Figura 33</b> Instalación de Open Virtual Switch.....	57
<b>Figura 34</b> Versión de OVS y versiones de OpenFlow soportadas por el mismo.....	57
<b>Figura 35</b> Instalación de paquete para reconocer adaptador USB2.0 a Ethernet.....	58
<b>Figura 36</b> Edición de archivo de configuración de red.....	59
<b>Figura 37</b> Interfaz eth1 habilitada.....	59
<b>Figura 38</b> Inicialización de OVS.....	60
<b>Figura 39</b> Creación de puente virtual con OVS.....	60
<b>Figura 40</b> Interfaces agregadas a puente OVS.....	61
<b>Figura 41</b> Habilitación y direccionamiento de interfaz bridge creada.....	61
<b>Figura 42</b> Habilitar comunicación de puente OVS con controlador mediante OpenFlow.....	62
<b>Figura 43</b> Ejecución de ejemplo para probar comunicación entre APs y controlador.....	62
<b>Figura 44</b> Verificación de comunicación entre los APs y el controlador.....	63
<b>Figura 45</b> Protocolo OpenFlow en ejecución de ejemplo.....	63
<b>Figura 46</b> Selección de protocolo inalámbrico y canal de funcionamiento.....	64
<b>Figura 47</b> Configuraciones de ESSID y conexión con bridge en la red.....	65
<b>Figura 48</b> Configuración de seguridad WPA2 en la red.....	65
<b>Figura 49</b> Configuración de firewall para admitir conexiones de estaciones a la red.....	66
<b>Figura 50</b> Comunicación con todos de los APs de la red desde API.....	67
<b>Figura 51</b> Análisis de canales en la banda 2,4 GHz.....	67
<b>Figura 52</b> Conexión a red SDWN desde smartphome Redmi Note 8.....	68
<b>Figura 53</b> Detalles de la conexión desde estación (STA).....	68
<b>Figura 54</b> Acceso a servidor VOD desde estación.....	69
<b>Figura 55</b> Funcionamiento de QoS sobre red SDWN.....	70
<b>Figura 56</b> Diagrama de flujo del funcionamiento de API RESTful para QoS.....	76
<b>Figura 57</b> Configuración de system-id y datapath-id en API.....	78
<b>Figura 58</b> Configuración de puerto para acceso a OVSDDB.....	78
<b>Figura 59</b> Modificación de ID en tablas de entrada de flujo.....	79
<b>Figura 60</b> Ejecución de programas desde controlador SDN.....	80
<b>Figura 61</b> Solicitud de conexión de controlador con OVSDDB de API.....	80
<b>Figura 62</b> Respuesta exitosa a conexión con OVSDDB.....	81
<b>Figura 63</b> Velocidad máxima de red inalámbrica en servidor Iperf3.....	82
<b>Figura 64</b> Configuración de colas mediante solicitudes POST.....	83
<b>Figura 65</b> Respuesta a solicitud POST en configuración de colas.....	84
<b>Figura 66</b> Visualización de colas configuradas desde navegador.....	84
<b>Figura 67</b> Configuración de marcaje mediante solicitudes POST.....	85
<b>Figura 68</b> Respuestas a solicitudes POST de marcaje.....	85
<b>Figura 69</b> Visualización de marcaje desde navegador.....	86
<b>Figura 70</b> Clasificación de tráfico mediante solicitudes POST.....	87
<b>Figura 71</b> Respuesta a solicitudes de clasificación de tráfico.....	87
<b>Figura 72</b> Visualización de clasificación de tráfico desde navegador.....	88
<b>Figura 73</b> Marcaje en paquetes SIP.....	90
<b>Figura 74</b> Marcaje en paquetes RTP.....	90
<b>Figura 75</b> Marcaje en paquetes FTP.....	91
<b>Figura 76</b> Marcaje en paquetes TCP debido al servicio de VOD.....	91
<b>Figura 77</b> Esquema de pruebas conforme a ITU-T Y.1540.....	93
<b>Figura 78</b> Evaluación de Ancho de banda en escenario 1, sin QoS.....	98



<b>Figura 79</b>	Evaluación de paquetes perdidos en escenario 1, sin QoS.....	99
<b>Figura 80</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	100
<b>Figura 81</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	100
<b>Figura 82</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	101
<b>Figura 83</b>	Evaluación de Tiempo de ida y vuelta en escenario 1, sin QoS.....	103
<b>Figura 84</b>	Evaluación de IPTD en escenario 1, sin QoS.....	104
<b>Figura 85</b>	Evaluación de Ancho de banda en escenario 2, sin QoS.....	105
<b>Figura 86</b>	Evaluación de Paquetes perdidos en escenario 2, sin QoS.....	106
<b>Figura 87</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	107
<b>Figura 88</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	107
<b>Figura 89</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	107
<b>Figura 90</b>	Evaluación de RTT en escenario 2, sin QoS. ....	109
<b>Figura 91</b>	Evaluación de IPTD en escenario 2, sin QoS.....	110
<b>Figura 92</b>	Evaluación de Ancho de banda en escenario 3, sin QoS.....	111
<b>Figura 93</b>	Evaluación de Paquetes perdidos en escenario 3, sin QoS.....	112
<b>Figura 94</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	113
<b>Figura 95</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	113
<b>Figura 96</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	113
<b>Figura 97</b>	Evaluación de RTT en escenario 3, sin QoS. ....	115
<b>Figura 98</b>	Evaluación de IPTD en escenario 3, sin QoS.....	116
<b>Figura 99</b>	Evaluación de Ancho de banda en escenario 4, sin QoS.....	117
<b>Figura 100</b>	Evaluación de Paquetes perdidos en escenario 4, sin QoS.....	118
<b>Figura 101</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	118
<b>Figura 102</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	119
<b>Figura 103</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	119
<b>Figura 104</b>	Evaluación de RTT en escenario 4, sin QoS. ....	121
<b>Figura 105</b>	Evaluación de IPTD en escenario 4, sin QoS.....	121
<b>Figura 106</b>	Evaluación de Ancho de banda en escenario 1, con QoS.....	123
<b>Figura 107</b>	Evaluación de paquetes perdidos en escenario 1, con QoS.....	124
<b>Figura 108</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	124
<b>Figura 109</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	125
<b>Figura 110</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	125
<b>Figura 111</b>	Evaluación de Tiempo de ida y vuelta en escenario 1, con QoS. ....	127
<b>Figura 112</b>	Evaluación de IPTD en escenario 1, con QoS.....	127
<b>Figura 113</b>	Evaluación de Ancho de banda en escenario 2, con QoS. ....	128
<b>Figura 114</b>	Evaluación de Paquetes perdidos en escenario 2, con QoS.....	129
<b>Figura 115</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	130
<b>Figura 116</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	130
<b>Figura 117</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	130
<b>Figura 118</b>	Evaluación de RTT en escenario 2, con QoS. ....	132
<b>Figura 119</b>	Evaluación de IPTD en escenario 2, con QoS.....	133
<b>Figura 120</b>	Evaluación de Ancho de banda en escenario 3, con QoS.....	134
<b>Figura 121</b>	Evaluación de paquetes perdidos en escenario 3, con QoS.....	135
<b>Figura 122</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	135
<b>Figura 123</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	136
<b>Figura 124</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	136

<b>Figura 125</b>	Evaluación de Tiempo de ida y vuelta en escenario 3, con QoS.....	138
<b>Figura 126</b>	Evaluación de IPTD en escenario 3, con QoS.....	138
<b>Figura 127</b>	Evaluación de Ancho de banda en escenario 4, con QoS.....	139
<b>Figura 128</b>	Evaluación de paquetes perdidos en escenario 4, con QoS.....	140
<b>Figura 129</b>	Cantidad de paquetes perdidos y enviados para servicio VoIP.....	141
<b>Figura 130</b>	Cantidad de paquetes perdidos y enviados para servicio VOD.....	141
<b>Figura 131</b>	Cantidad de paquetes perdidos y enviados para servicio FTP.....	141
<b>Figura 132</b>	Evaluación de Tiempo de ida y vuelta en escenario 4, con QoS.....	143
<b>Figura 133</b>	Evaluación de IPTD en escenario 4, con QoS.....	143
<b>Figura 134</b>	Comparación de AB en escenarios evaluados antes y después de aplicar QoS.	146
<b>Figura 135</b>	Comparación de paquetes perdidos en escenarios evaluados sin y con QoS.....	148
<b>Figura 136</b>	Comparación de RTT en escenarios evaluados antes y después de QoS. ....	150
<b>Figura 137</b>	Comparación de IPTD en escenarios evaluados antes y después de QoS.....	152
<b>Figura 138</b>	Instalación de dependencias para servicio Jellyfin.....	163
<b>Figura 139</b>	Descarga de firma GPG del equipo de Jellyfin. ....	164
<b>Figura 140</b>	Edición de archivo jellyfin.sources. ....	164
<b>Figura 141</b>	Instalación de Jellyfin.....	165
<b>Figura 142</b>	Administración de servicio Jellyfin.....	165
<b>Figura 143</b>	Creación de usuario para acceder a interfaz web de Jellyfin. ....	166
<b>Figura 144</b>	Inicio de sesión en GUI web de Jellyfin.....	166
<b>Figura 145</b>	Adición de contenido multimedia a servidor.....	167
<b>Figura 146</b>	Visualización de contenido multimedia.....	168
<b>Figura 147</b>	Instalación de servicio vsftpd. ....	169
<b>Figura 148</b>	Administración del servicio.....	169
<b>Figura 149</b>	Aplicación de configuraciones realizadas. ....	170
<b>Figura 150</b>	Creación de usuario para acceso a FTP. ....	171
<b>Figura 151</b>	Adición de usuario FTP al archivo userlist.....	171
<b>Figura 152</b>	Creación de directorio compartido con FTP.....	171
<b>Figura 153</b>	Acceso a servidor FTP median agente FileZilla.....	172
<b>Figura 154</b>	Descarga de máquina virtual Issabel. ....	173
<b>Figura 155</b>	Usuarios y extensiones creadas. ....	174
<b>Figura 156</b>	Funcionamiento de servicio VoIP. ....	174

## ÍNDICE DE TABLAS

<b>Tabla 1</b>	Comparación de estándares IEEE 802.11. ....	14
<b>Tabla 2</b>	Clasificación y Priorización de Tráfico. ....	28
<b>Tabla 3</b>	Características de los principales servicios multimedia. ....	29
<b>Tabla 4</b>	Siglas y representación de componentes de red definidos en ITU-T Y.1540.....	32
<b>Tabla 5</b>	Definición de clases de QoS y objetivos de rendimiento de la red. ....	38
<b>Tabla 6</b>	Asociación de clases de QoS de UIT-T Y.1541 con PHB de DiffServ.....	38
<b>Tabla 7</b>	Comparación de controladores SDN.....	42
<b>Tabla 8</b>	Características de prototipo SDWN definido por Moncayo (2023).....	46
<b>Tabla 9</b>	Requerimientos de máquina virtual para controlador. ....	48

<b>Tabla 10</b>	Prerrequisitos para la instalación de Ryu. ....	49
<b>Tabla 11</b>	Distribución de canales inalámbricos en APs.....	64
<b>Tabla 12</b>	Puertos de funcionamiento y prioridades en servicios de red SDWN. ....	71
<b>Tabla 13</b>	Puntos finales para configuración de procesos de QoS.....	72
<b>Tabla 14</b>	Formato JSON para las solicitudes de los procesos de QoS.....	73
<b>Tabla 15</b>	Ejemplos de solicitudes a los puntos finales definidos para QoS. ....	74
<b>Tabla 16</b>	Distribución de system-id y datapath-id en cada AP. ....	78
<b>Tabla 17</b>	Parámetros a considerar para configuración de colas. ....	83
<b>Tabla 18</b>	Parámetros de marcaje en la red.....	85
<b>Tabla 19</b>	Clasificación de tráfico en las colas creadas. ....	87
<b>Tabla 20</b>	Herramientas de medición seleccionadas.....	94
<b>Tabla 21</b>	Dirección IP y puerto de servicios para pruebas. ....	95
<b>Tabla 22</b>	Descripción de escenarios para evaluación de red. ....	96
<b>Tabla 23</b>	Datos de IPLR obtenidos en el Escenario 1, sin QoS ....	101
<b>Tabla 24</b>	Disponibilidad de red en Escenario 1, sin QoS.....	102
<b>Tabla 25</b>	Datos de IPLR obtenidos en el Escenario 2, sin QoS. ....	108
<b>Tabla 26</b>	Disponibilidad de red en Escenario 2, sin QoS.....	108
<b>Tabla 27</b>	Datos de IPLR obtenidos en el Escenario 3, sin QoS. ....	114
<b>Tabla 28</b>	Disponibilidad de red en Escenario 3, sin QoS.....	114
<b>Tabla 29</b>	Datos de IPLR obtenidos en el Escenario 4, sin QoS. ....	119
<b>Tabla 30</b>	Disponibilidad de red en Escenario 4, sin QoS.....	120
<b>Tabla 31</b>	Datos de IPLR obtenidos en el Escenario 1, con QoS. ....	125
<b>Tabla 32</b>	Disponibilidad de red en Escenario 1, con QoS.....	126
<b>Tabla 33</b>	Datos de IPLR obtenidos en el Escenario 2, con QoS. ....	131
<b>Tabla 34</b>	Disponibilidad de red en Escenario 2, con QoS.....	131
<b>Tabla 35</b>	Datos de IPLR obtenidos en el Escenario 3, con QoS. ....	136
<b>Tabla 36</b>	Disponibilidad de red en Escenario 3, con QoS.....	137
<b>Tabla 37</b>	Datos de IPLR obtenidos en el Escenario 4, con QoS. ....	142
<b>Tabla 38</b>	Disponibilidad de red en Escenario 4, con QoS.....	142
<b>Tabla 39</b>	Ancho de banda promedio en escenarios de pruebas con y sin QoS. ....	146
<b>Tabla 40</b>	Tasa de paquetes perdidos en escenarios de pruebas con y sin QoS.....	148
<b>Tabla 41</b>	Disponibilidad de servicios en escenarios de pruebas con y sin QoS.....	149
<b>Tabla 42</b>	RTT promedio en escenarios de pruebas con y sin QoS. ....	150
<b>Tabla 43</b>	IPTD promedio en escenarios de pruebas con y sin QoS. ....	152
<b>Tabla 44</b>	Características de máquinas alojadas en VirtualBox.....	162
<b>Tabla 45</b>	Prerrequisitos para la instalación de Jellyfin.....	163
<b>Tabla 46</b>	Configuración de seguridad y acceso al servidor FTP. ....	170

## LISTA DE SIGLAS

- AB.** Ancho de Banda
- AF.** Assured Forwarding
- AP.** Access Point
- API.** Application Programming Interface
- BE.** Best Effort
- CS.** Class Selector
- CURL.** Cliente URL
- DiffServ.** Servicios Diferenciados
- DP.** Drop Probability
- DSCP.** DiffServ Code Point
- EF.** Expedited Forwarding
- ESS.** Extended Service Set
- ESSID.** Extended Service Set Identifier
- FTP.** File Transfer Protocol
- HTB.** Hierarchical Token Bucket
- HTTP.** Hypertext Transfer Protocol
- IEEE.** Institute of Electrical and Electronics Engineers
- IETF.** Internet Engineering Task Force
- IntServ.** Servicios Integrados
- IP ENC.** IP Explicit Congestion Notification
- IP.** Internet Protocol
- IPLR.** Tasa de pérdida de paquetes IP
- IPP.** IP Precedence
- IPTD.** Retardo de transferencia de paquetes IP
- JSON.** JavaScript Object Notation
- KPI.** Key Performance Indicator
- MAC.** Media Access Control Layer
- MSDU.** MAC Services Data Unit
- ONF.** Open Networking Foundation

**OVS.** Open Virtual Switch

**OVSDB.** Open Virtual Switch Database

**PDU.** Protocol Data Unit

**PHB.** Per-Hop Behavior

**PHY.** Physical Layer

**PPDIOO.** Preparar, Planificar, Diseñar, Implementar, Operar y Optimizar

**QoS.** Quality of Services

**REST.** Representational State Transfer

**RPi.** Raspberry Pi

**RSVP.** Resource Reservation Protocol

**RTP.** Real Time Protocol

**RTT.** Round-Trip Time

**SDN.** Software Defined Networking

**SDWLAN.** Software Defined Wireless Local Area Network

**SDWN.** Software Defined Wireless Network

**SIP.** Session Initiation Protocol

**SSH.** Secure Shell

**SSID.** Service Set Identifier

**STA.** Station

**TC.** Traffic Class

**TCP.** Transmission Control Protocol

**ToS.** Type of Service

**UDP.** User Datagram Protocol

**URL.** Uniform Resource Locator

**VOD.** Video on Demand

**VoIP.** Voice over IP

**WiFi.** Wireless fidelity

**WLAN.** Wireless Local Area Network

## RESUMEN EJECUTIVO

El presente trabajo de titulación se centra en la aplicación y evaluación de Calidad de Servicio (QoS) sobre una Red Inalámbrica Definida por Software (SDWN), fundamentado en la creciente demanda de redes eficientes ante el aumento de servicios multimedia que cada vez exigen mayores requisitos de calidad para su funcionamiento. El objetivo principal es evaluar el desempeño de un prototipo SDWN configurado con QoS, tomando como referencia parámetros de rendimiento definidos en la recomendación ITU-T Y.1540. La metodología empleada incluye una serie de fases: un análisis teórico sobre la gestión de QoS en entornos SDN; la implementación de un prototipo SDWN utilizando el controlador Ryu, puntos de acceso con soporte para OpenFlow, y la integración de servicios multimedia que incluyen VoIP, VOD y FTP, priorizando el tráfico mediante el modelo de servicios diferenciados (DiffServ); y, finalmente, se valida el rendimiento del sistema en función de escenarios relevantes alineados con las directrices de ITU-T Y.1540. Los resultados muestran mejoras significativas al implementar QoS, incluyendo una reducción en la pérdida de paquetes, mayor disponibilidad de servicios, menor tiempo de ida y vuelta (RTT), latencia reducida, y eficiencia en la asignación de ancho de banda. Estos hallazgos validan el impacto positivo de QoS, particularmente en condiciones de alto tráfico de red. En conclusión, el enfoque de gestión centralizada de los entornos SDN para la optimización de QoS proporciona una solución eficaz para redes que requieren una calidad de servicio rigurosa, a través de una configuración y administración simplificadas.

**Palabras clave:** DiffServ, OpenFlow, QoS, Ryu, SDWN.

## ABSTRACT

The present project is oriented towards the implementation and evaluation of Quality of Service (QoS) in a Software-Defined Wireless Network (SDWN). This initiative is driven by increasing demand for efficient networks capable of supporting bandwidth-intensive multimedia services. This work aims to assess the performance of an SDWN prototype configured with QoS, based on network performance parameters defined by the International Telecommunication Union's ITU-T Y.1540 standard. The methodology encompasses several phases: initially, a theoretical analysis of QoS management in SDN environments; followed by the implementation of an SDWN using the Ryu controller, OpenFlow-supported access points, and multimedia services such as VoIP, VOD, and FTP, prioritizing traffic with the Differentiated Services (DiffServ) model; and performance verification through relevant scenarios aligned with the ITU-T Y.1540 guidelines. The results of this study demonstrate significant improvements in various metrics following the implementation of QoS, including a reduction in packet loss rate, enhancement of network availability, decrease in round-trip time, reduction in transmission delay, and improvement in bandwidth allocation efficiency. These outcomes validate the positive impact of QoS, particularly under conditions of high network traffic. In conclusion, the centralized management approach of SDN environments for QoS optimization provides an effective solution for networks requiring stringent quality of service, through streamlined configuration and administration.

**Keywords:** DiffServ, OpenFlow, QoS, Ryu, SDWN.

## 1. CAPÍTULO I: Antecedentes

En este capítulo se presenta el planteamiento general del proyecto, incluyendo la problemática, los objetivos, el alcance y la justificación para la realización del trabajo.

### 1.1.Tema

Aplicación y Evaluación de Calidad de Servicio (QoS) sobre una Red Inalámbrica Definida por Software (SDWN).

### 1.2.Problema

De acuerdo con Shin et al. (2012), las redes de comunicación fundamentaron sus bases en la década de 1970. Aunque han surgido diversos paradigmas enfocados en mejorar su eficiencia, todos se encontraban sujetos al modelo inicial donde los dispositivos de red como enrutadores y switches carecían de inteligencia dado que limitaban sus funciones a la lectura de direcciones y transferencia de paquetes al sistema (Murcia & Beltrán, 2021), además estas funciones dependían directamente de las características del hardware, lo cual constituyó la principal limitación en estas redes. No obstante, es importante resaltar que la adopción de las redes tradicionales ha permitido optimizar numerosas actividades cotidianas, de esa forma han dado lugar a una amplia variedad de servicios en red con distintas funciones. En consecuencia, dicho suceso ha provocado el incremento de dispositivos conectados a la red, lo que ha generado una mayor demanda y, a su vez, ha aumentado la complejidad en la administración.

Las soluciones basadas en cables eran las tecnologías predominantes en el ámbito de las redes. Sin embargo, la implementación de los estándares IEEE 802.11 ha generado un impacto significativo en el mercado debido a que se caracterizan por establecer conexiones de red sin la necesidad de cables físicos lo cual reduce significativamente los costes de infraestructura (Banerji & Chowdhury, 2013). Si bien al principio este estándar ofrecía una velocidad de transmisión de datos de 2 Mbps, esto fue aumentando en versiones posteriores,



llegando hasta lo que hoy se conoce como IEEE 802.11ax o WiFi 6 con velocidades de hasta 9,6 Gbps (en condiciones ideales) lo cual da paso a la proliferación de diversas aplicaciones sobre estas redes (IEEE SA, 2023), a pesar de ello, como en toda tecnología, las redes inalámbricas no son perfectas y se enfrentan a muchos problemas como seguridad, alteraciones en el canal de comunicación, entre otros.

Por tanto, en base a lo presentado anteriormente es posible notar que en la actualidad el principal problema que enfrentan las redes es la demanda causada tanto por el número de usuarios en red como de servicios, en vista de ello es inevitable imaginar la congestión de red en escenarios con alta demanda de recursos. En cierta medida, el tráfico de una red es diverso en el sentido de que diferentes servicios funcionan bajo ciertos requerimientos de red, es así como resalta la Calidad de Servicio (Quality of Service, QoS), la cual se encarga de priorizar el tráfico en base a sus requerimientos, de forma que se permita reducir la congestión y así garantizar una entrega confiable de los paquetes de datos (Schmitt & Wolf, 1997).

Ahora, es importante analizar que las nuevas aplicaciones cada vez tienden a precisar de mayores requerimientos de red. A esto los autores Chen et al. (2017) señalan que existe una previsión de que la demanda futura de servicios inalámbricos excederá ampliamente la capacidad de las redes de área local existentes, de modo que la estructura de las redes tradicionales resultará poco eficiente para estos escenarios. Por otro lado, en los últimos años se ha venido utilizando el concepto de redes definidas por software (Software Defined Network, SDN), mismas que se caracterizan por el control centralizado de la red, además de poseer ventajas como; programabilidad y flexibilidad de red, robustez, neutralización de proveedores y optimización global (Macedo et al., 2015). Bajo este mismo paradigma se ha llevado a cabo la integración entre SDN y tecnologías inalámbricas, dando lugar a las redes inalámbricas definidas por software (Software Defined Wireless Network, SDWN).

Debido a su condición como una tecnología en desarrollo, SDWN todavía debe superar numerosos desafíos. En este contexto, Moncayo (2023) destaca que uno de los principales retos recae sobre los administradores de redes dado que únicamente se encuentra familiarizados con redes convencionales. Por lo tanto, debe consolidar los conocimientos necesarios para llevar a cabo procesos de administración y operación sobre esta nueva tecnología. Por otro lado, otro reto importante radica en la exploración de esta tecnología, de modo que se pueda llevar a cabo la consolidación de protocolos y técnicas de administración de red convencionales (Chen et al., 2017). En otras palabras, es preciso comprender cómo administrar los procesos necesarios de una red en la nueva estructura que presenta las redes SDWN, de forma que esta tecnología sea desplegada en entornos que la requieran y que, a su vez, el personal encargado de su administración adquiera las cualidades necesarias para trabajar en este escenario.

Finalmente, desplegar calidad de servicio en arquitecturas SDWN no es fácil debido a que difieren en muchos aspectos de las redes tradicionales, mismos que deben ser evaluados de manera exhaustiva considerando principalmente los prototipos de redes SDWN existentes con lo cual se identifique sus características, a fin establecer un conocimiento global, lo cual ayudará a fundamentar las bases necesarias para tal implementación. Asimismo, la evaluación de este método permitirá determinar qué configuraciones pueden ser adecuadas en determinados entornos.

### **1.3. Objetivos**

#### ***1.3.1. Objetivo general***

Evaluar la calidad de servicio en redes SDWN mediante la identificación y medición de métricas basadas en el estándar ITU-T Y.1540.

#### ***1.3.2. Objetivos específicos***

- Estudiar el fundamento teórico que sustenta la tecnología SDWN mediante una investigación bibliográfica, estableciendo una línea base de conocimientos para determinar los mecanismos de calidad de servicio que pueden ser implementadas en este paradigma.
- Desplegar un modelo de simulación o un escenario de prueba para la implementación de calidad de servicio sobre una arquitectura de red SDWN con base en la recomendación ITU-T Y.1541, donde se priorice el tráfico de datos en función de un escenario de servicios multimedia previamente instaurados.
- Recopilar datos para medir y evaluar la metodología de QoS implementada en la red SDN en un entorno inalámbrico, considerando los parámetros de rendimiento más sobresalientes de la red referidos en el estándar ITU-T Y.1540.

### **1.4. Alcance**

El propósito de este proyecto radica en el análisis de la calidad de servicio sobre una arquitectura de red inalámbrica definida por software, con la finalidad de conocer cómo se realiza la configuración de este método y qué mecanismos pueden ser aplicados sobre esta tecnología, además evaluar su funcionalidad y rendimiento, para lo cual se examinará los diferentes parámetros asociados con el desempeño de la red en la ejecución de servicios multimedia de baja, media y alta prioridad, como lo son FTP, Streaming de video almacenado y VoIP respectivamente, donde se tomará en consideración el estándar ITU-T Y.1540, mismo que define parámetros de rendimiento para redes IP. Para el desarrollo de este proyecto se

plantean tres fases: análisis, implementación y verificación, las cuales se detallan a continuación.

La fase de análisis abarca el estudio de las características que posee la tecnología SDWN, su enfoque hacia la calidad de servicio y las técnicas que se aplican sobre esta arquitectura. Además, se estudiará los servicios multimedia antes definidos, en función de la recomendación ITU-T Y.1541. Así también se estudiará la recomendación ITU-T Y.1540 que define la medición de parámetros cuantitativos del rendimiento de la red. Y finalmente, se analizará el estándar IEEE 802.11n debido a que fundamenta la comunicación inalámbrica de la red propuesta.

Posteriormente la fase de implementación se llevará a cabo en tres etapas, tales como:

- i) establecimiento de la red física SDWN donde se determinará los protocolo IEEE 802.11n, OpenFlow y la API RESTful, para comunicación inalámbrica, comunicación del controlador con los puntos de acceso y comunicación del controlador con capa aplicación respectivamente; topología lógica que constituye el direccionamiento IPv4 y configuración del controlador Ryu, los cuales consolidan la arquitectura, para lo cual, cabe señalar que la topología de red a implementarse toma como línea base el trabajo de Moncayo (2023), incrementando el número de puntos de acceso, estaciones móviles y desplegando servidores de aplicaciones para generar un flujo<sup>1</sup> de tráfico diverso y que permita el estudio de QoS en redes SDWN, tal como se muestra en la **Figura 1**; ii) implementación de un entorno de servidores multimedia con los servicios FTP, Streaming de video almacenado y VoIP, los cuales generan, dependiendo de la aplicación, diferentes clases de tráfico; lo cual será insumo para establecer y analizar políticas de QoS en una red SDWN; iii) configuración de QoS en la red SDWN donde se empleará la arquitectura API REST para acceder y configurar las funciones de QoS en el controlador Ruy

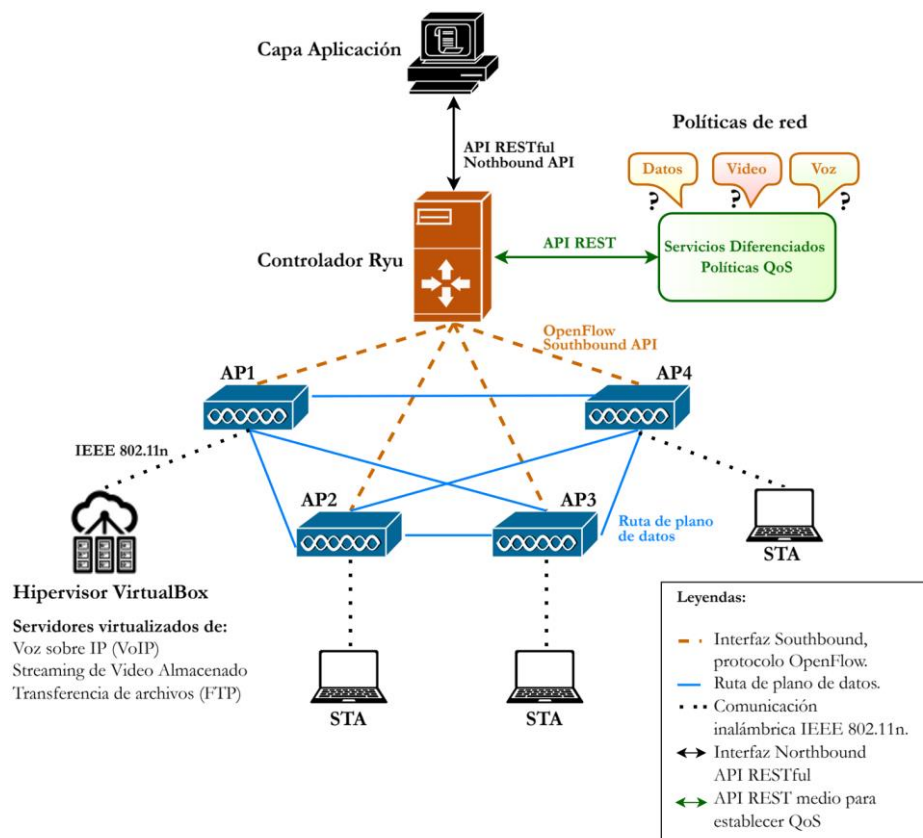
---

<sup>1</sup> Un flujo es una secuencia de paquetes con una dirección de origen, dirección de destino y número de puerto comunes (Malik et al., 2015) (Malik et al., 2015).

y así establecer el mecanismo de Servicios Diferenciados (DiffServ), con base en la recomendación ITU-T Y.1541, la cual permite analizar el tipo de tráfico de red, y en función de ello clasificarlo por clases y proporcionarle el tratamiento sugerido por dicha recomendación.

Finalmente, considerando las ventajas que brindan las redes SDN en general, en la fase de verificación se plantea realizar un análisis de los parámetros de rendimiento de red basados en la recomendación ITU-T Y.1540, misma que determina donde realizar las mediciones y los indicadores de red que pueden ser medidos cuantitativamente para conocer sus variaciones y la tendencia que presentan luego de haber sido aplicados calidad de servicio. Con las mediciones realizadas, se podrá estudiar y analizar el comportamiento y rendimiento del mecanismo de calidad de servicio sobre la arquitectura SDWN propuesta.

**Figura 1**  
*Topología de red física SDWN.*



*Nota.* Adoptado de Moncayo (2023).

### **1.5. Justificación**

El planteamiento de este proyecto surge debido a la creciente demanda en las redes de comunicación a lo largo de los años, en el mismo contexto, Cisco Systems (2021) indica que el tráfico mensual por habitante ha experimentado un incremento significativo, variando desde 12,9 GB hasta 35,5 GB entre los años 2016 y 2021, lo cual ha generado dificultades en la administración de las redes convencionales. Este hecho evidencia la necesidad de establecer criterios de administración de redes que permitan hacer frente a estos desafíos. En este sentido, es relevante considerar las ventajas que brinda la tecnología SDN, del mismo modo, estudiar su principio de funcionamiento, así como comprender los retos que pueden surgir en el despliegue de este paradigma.

Además, Gonzalez (2020) manifiesta que las compañías que ofrecen servicios de telecomunicaciones destinan una parte significativa de sus ingresos a procesos de operación y mantenimiento de sus infraestructuras. El mismo autor resalta que, las redes de datos convencionales presentan costos elevados en términos de mantenimiento debido a su modelo tecnológico inflexible. Así, considerando que, dentro del marco de las redes SDN, los protocolos, estándares y demás, se encuentran en constante evolución, posibilitan la puesta en marcha de redes con mayor flexibilidad y automatización, de modo que un estudio de los mismos permitirá evaluar su factibilidad, al mismo tiempo que contribuye en la investigación de este paradigma.

Una de las características de las redes definidas por software es simplificar el control y la gestión del tráfico de la red, permitiendo la programación de componentes de red y brindando la capacidad de innovación, con ello, busca superar las limitaciones de las arquitecturas y controles tradicionales, ofreciendo métodos de gestión que promueven flexibilidad, mayor control y la posibilidad de introducir modelos innovadores (Albu-Salih, 2022). A ello, Gonzalez (2020) menciona que, este nuevo paradigma podría proveer la implementación de

mecanismos de QoS en tiempo real, de modo que la reserva de recursos de ancho de banda en las interfaces de los equipamientos de red podrá estar disponible en el momento en el que es requerido por una aplicación, optimizando la distribución y el uso del ancho de banda total de la red.

Finalmente, es importante destacar que el objetivo nueve, definido en los objetivos de desarrollo sostenible (ODS) establecidos por el Programa para las Naciones Unidas de Desarrollo (2023), establece “Construir infraestructuras resilientes, promover la industrialización y fomentar la innovación”. De este modo, haciendo énfasis en el término resiliente, es importante considerar que las redes de comunicación deben ser capaces de adaptarse a los nuevos enfoques de comunicaciones y del mismo modo, brindar un servicio adecuado a sus usuarios.

## 2. CAPÍTULO II: Estado del arte

Este capítulo abarca la fundamentación teórica necesaria para el desarrollo del proyecto, la cual toma como punto de partida el estudio del protocolo inalámbrico IEEE 802.11n mismo que fue planteado en la arquitectura de red. Luego, se lleva a cabo una investigación de la Calidad de Servicio (QoS), considerando; métricas, modelos, técnicas, y normativas. Finalmente se presenta las Redes Definidas por Software (SDWN), donde se toma en cuenta; arquitectura, estandarización, protocolos, componentes de red y esencialmente la gestión e integración de QoS sobre este paradigma.

### 2.1. Estándar IEEE 802.11n

De acuerdo con Stallings (2014), en 1990 el grupo de trabajo IEEE 802 formó IEEE 802.11 enfocado exclusivamente en las Redes Inalámbricas de Área Local (WLAN, Wireless Local Area Network), donde el objetivo principal fue desarrollar protocolos y especificaciones de las capas MAC y Física que permitieran establecer las redes WLAN y hacer frente a la demanda de tal tecnología, es así como se han establecido algunos estándares enfocados en este propósito, tal como se muestra en la **Tabla 1**. Además, es importante destacar que todos los estándares IEEE 802.11 están diseñados de forma que sean compatibles con versiones anteriores (IEEE SA, 2023). Sin embargo, en este estudio resalta el protocolo 802.11n también conocido como Wi-Fi 4, y aunque a la fecha no es el protocolo más actual, aún sigue siendo ampliamente utilizado.

El estándar IEEE 802.11n fue establecido en el 2009, no obstante, desde el año 2007 los fabricantes comenzaron a distribuir dispositivos compatibles con este estándar, mismos que se basaban en el borrador 2.0 debido a que era considerado estable para la comercialización de productos (Hajlaoui & Jabri, 2012). El objetivo principal de 802.11n fue aumentar la velocidad



efectiva de la red a más de 100 Mbps<sup>2</sup>, lo cual iba más allá de considerar el esquema de codificación de la señal, y entraron en juego la arquitectura de antenas y la estructura de las tramas MAC (Stallings, 2014). Finalmente, una característica por resaltar de este protocolo es que funciona en las bandas de 2.4 y 5 GHz<sup>3</sup>.

**Tabla 1**

*Comparación de estándares IEEE 802.11.*

Estándar	Año de despliegue	Velocidad máxima	Banda de frecuencia	Ancho de banda	Modulación de orden más alto	Uso de espectro
802.11	1997	1 a 2 Mbps	2.4 GHz	20 MHz	QPSK	DSSS
802.11b	1999	11 Mbps	2.4 GHz	20 MHz	11 CCK	OFDM
802.11a	1999	54 Mbps	5 GHz	20 MHz	64 QAM	DSSS
802.11g	2003	54 Mbps	2.4 GHz	20 MHz	64 QAM	DSSS, OFDM
802.11n	2009	65 a 600 Mbps	2.4/5 GHz	20/40 MHz	64 QAM	OFDM
802.11ac	2013	78 Mbps a 3.2 Gbps	5 GHz	40/80/160 MHz	256 QAM	OFDM
802.11ax	2021	9.6 Gbps	2.4/5 GHz	20/40/80/160MHz	256 QAM	OFDMA

*Nota.* Adoptado de Stallings (2014) e IEEE SA (2023). Esta tabla muestra la evolución de los estándares 802.11, donde se presentan características clave de cada estándar, además se incluyen términos como; QPSK, QAM, CCK, DSSS, OFDM y OFDMA, las cuales son técnicas se utilizan para modular y transmitir señales en sistemas de comunicación inalámbrica, cada una con características y aplicaciones específicas.

### 2.1.1. Capa Control de Acceso al Medio (MAC)

La subcapa MAC, que forma parte de la capa enlace de datos en la arquitectura TCP/IP, actúa como una interfaz entre la subcapa de Control de Enlace Lógico (LLC, Logical Link Control) y la capa física. Entre sus principales funciones se incluyen el envío y la recepción de tramas, la sincronización, el control de errores, el formato de tramas, así como la fragmentación

<sup>2</sup> Mbps es Megabits por segundo. “Mega” es un prefijo métrico, mientras que “bits por segundo” es la unidad que expresa la velocidad de transferencia de datos, e indica la cantidad de datos que pueden ser transmitidos en un segundo.

<sup>3</sup> GHz es Gigahertz. “Giga” es un prefijo métrico y “Hertz” es la unidad para medir la frecuencia, la cual hace referencia al número de repeticiones de un fenómeno periódico en un intervalo de tiempo específico.

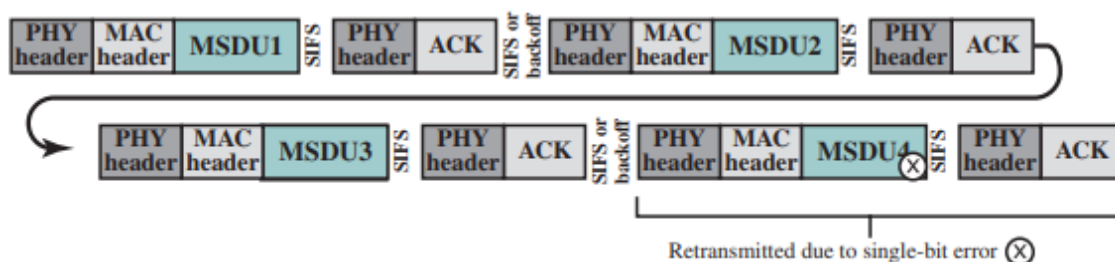
y reensamble de las mismas (Priya & Singh, 2016). En este contexto, las mejoras presentadas por 802.11n son las siguientes.

### 2.1.1.1. Agregación de tramas.

Esta optimización permite concatenar múltiples tramas MAC en un solo bloque para su posterior transmisión, con el objetivo de mejorar la eficiencia del enlace inalámbrico (IEEE, 2009). Así, Stallings (2014) indica que una vez que la estación ha obtenido el medio para transmitir, la agregación de tramas permite enviar largos paquetes sin las demoras significativas que conlleva el envío individual de tramas (ver en **Figura 2**), además, el encabezado de capa física se envía solo al comienzo de la trama agregada, y de manera similar, el receptor envía únicamente un bloque de acuse de recibo o ACK (Acknowledgement).

**Figura 2**

*Transmisión sin agregación de tramas.*



*Nota.* Tomado de Stallings (2014). Esta figura muestra la transmisión individual de tramas, así, cada conjunto de datos (MSDU) cuenta con sus cabeceras, además, por cada trama se establece un ACK, y también, todas ellas agregan el intervalo de tiempo Short Interface Space (SIFS) utilizado para controlar el acceso al medio. En algunos casos, se añade el período Backoff de duración aleatoria para evitar colisiones cuando se detecta que el medio está ocupado.

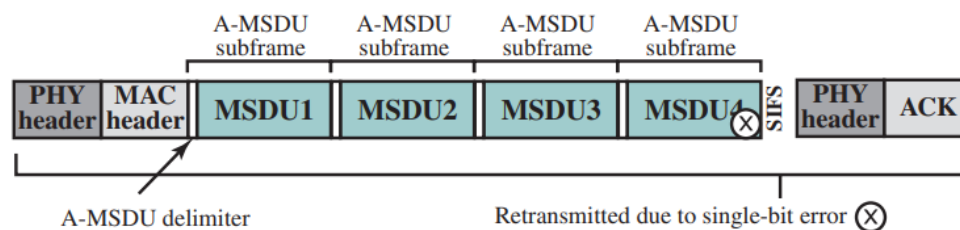
Existen tres mecanismos de agregación de tramas; Agregación de Unidad de Datos de Servicio MAC (A-MSDU, Aggregation MAC Services Data Unit), Agregación de Unidad de Datos de Protocolo MAC (A-MPDU, Aggregation MAC Protocol Data Unit) y la combinación de ambos. Así, los dos primeros mecanismos se consideran como agregación de tramas de un solo nivel mientras que el tercero es de dos niveles (Visoottiviseth et al., 2009).

A-MSDU permite que varios conjuntos de datos o también conocidos como carga útil (MSDU) se envíen al mismo receptor agrupados en un solo PDU de capa MAC (MPDU, MAC Protocol Data Unit) (ver en **Figura 3**). Es decir que existirá una única cabecera MAC, sin embargo, si existe un error en la transmisión es necesario retransmitir toda la trama agregada (Stallings, 2014). Además, Hajlaoui & Jabri (2012) indican que el tamaño máximo de una trama A-MSDU es de 8192 bytes y los receptores 802.11n pueden confirmar una trama A-MSDU enviando un solo ACK.

Visoottiviseth et al. (2009) manifiestan que A-MPDU combina múltiples PDU de capa MAC (MPDU), y este método ocurre después de que se han establecido las cabeceras MAC para cada MSDU. De ese modo, se obtiene la carga adicional de dichas cabeceras. No obstante, la trama agrupada posee solo una cabecera de capa física (ver en **Figura 4**). El tamaño máximo es de 65535 bytes, y a diferencia de A-MSDU, cuando existe un error de transmisión, es posible retransmitir únicamente la trama perdida y no toda la trama agregada (Hajlaoui & Jabri, 2012).

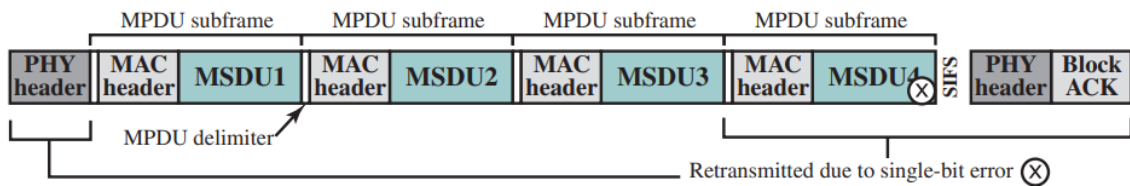
Por su parte, la combinación de los dos mecanismos establece que un MPDU puede contener algunos MSDU, y a su vez, una trama agrupada puede contener varios MPDUs agregados, tal como se muestra en la **Figura 5** (Stallings, 2014).

**Figura 3**  
*Agregación de tramas A-MSDU.*



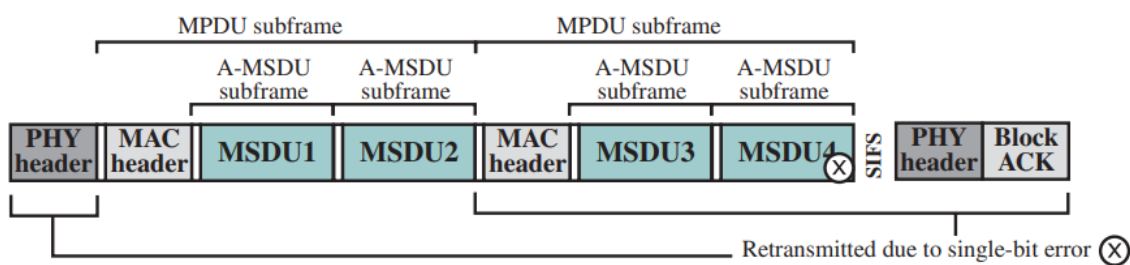
*Nota.* Tomado de Stallings (2014).

**Figura 4**  
Agregación de tramas A-MPDU.



*Nota.* Tomado de Stallings (2014).

**Figura 5**  
Agregación A-MSDU y A-MPDU combinadas.



*Nota.* Tomado de Stallings (2014).

### 2.1.2. Capa Física (PHY)

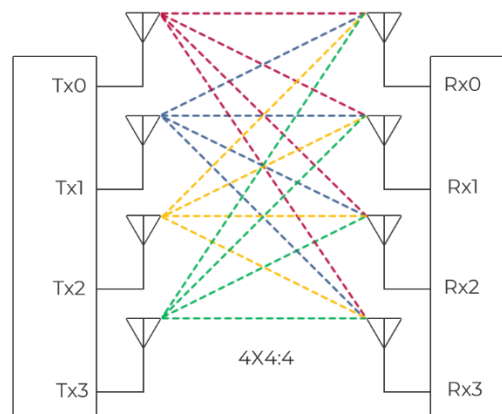
La capa física (PHY) es responsable de la transmisión de bits a través del medio físico. Además, se encarga de establecer la conexión entre los dispositivos y medios de comunicación, ya sea en configuraciones punto a punto o multipunto (Priya & Singh, 2016). Asimismo, define los esquemas de codificación que determinan la representación de los bits como señales eléctricas o electromagnéticas (Geeks for Geeks, 2020).

En el contexto de las mejoras presentadas por el estándar 802.11n, Stallings (2014) señala la integración de la tecnología MIMO (Multiple Input Multiple Output), la cual permite la transmisión y recepción simultánea a través de múltiples antenas (ver **Figura 6**), en consecuencia, esta innovación ha posibilitado el aumento significativo de las tasas de transferencia de datos, alcanzando velocidades de hasta 600 Mbps.

### 2.1.2.1.MIMO.

Esta tecnología es utilizada en comunicaciones inalámbricas para mejorar el rendimiento de los sistemas de transmisión y recepción, debido a que emplea la técnica de multiplexación espacial para transmitir dos o más flujos de datos paralelos en el mismo canal de frecuencia, además, un sistema MIMO se compone por un número de transmisores (N) y receptores (M), mismo que se representa como NxM (Nabar, 2014). El estándar 802.11n adopta las configuraciones que van desde 1x1 hasta 4x4, mismas que permiten una variedad de opciones para adaptarse a diferentes necesidades de rendimiento y cobertura en redes inalámbricas (Hajlaoui & Jabri, 2012).

**Figura 6**  
*Sistema MIMO de 4x4.*



*Nota.* Tomado de Powertec Wireless Technology (2023).

### 2.2.Calidad de servicio (QoS)

La calidad de servicio (QoS, Quality of Service) es un concepto crucial en el ámbito de las redes de comunicación. A medida que estas redes se expanden, se vuelven esenciales en diversas actividades cotidianas, por ello es fundamental garantizar que los servicios ofrecidos sean confiables y eficientes. Sin embargo, dicha expansión ha dado lugar al surgimiento de nuevas aplicaciones y servicios, desencadenando un fuerte aumento en el tráfico de red, lo cual

genera congestión<sup>4</sup>, degradación en la calidad del servicio o incluso la interrupción del mismo. Y aunque aumentar el ancho de banda podría considerarse una solución, esta opción suele ser costosa, por consiguiente, una alternativa más rentable es implementar una política de garantía para controlar la congestión del tráfico (Huawei, 2023).

Además, es importante reconocer que las aplicaciones generan flujos de tráfico distintos, y requieren un tratamiento específico para garantizar una transmisión exitosa (Karakus & Durrezi, 2017). Así, QoS es un conjunto de políticas que brindan la capacidad de gestionar flujos específicos, permitiendo determinar el orden en que se manejan los paquetes y asegurar la distribución adecuada de ancho de banda a cada aplicación o flujo de datos (Fortinet, 2022).

Por otro lado, la Unión Internacional de Telecomunicaciones (UIT) define la QoS como “La totalidad de las características de un servicio de telecomunicaciones que determinan su capacidad para satisfacer las necesidades explícitas e implícitas del usuario del servicio” (ITU-T E.800, 2008). Esta definición abarca tanto la calidad técnica proporcionada por los equipos de red como la calidad percibida por el usuario, que se conoce como Calidad de Experiencia<sup>5</sup> (QoE, Quality of Experience).

La diferencia entre QoS y QoE radica en su enfoque y naturaleza de evaluación. La primera se centra en parámetros técnicos de la red, los mismos que pueden ser medidos cuantitativamente, mientras que la segunda se relaciona con la percepción subjetiva del usuario, es decir indicadores cualitativos (ITU-T E.800, 2008). Estos parámetros de evaluación, comúnmente denominados como Indicadores Clave de Rendimiento (KPI, Key Performance

---

<sup>4</sup> La congestión se produce cuando el tráfico excede la capacidad máxima de la red, lo que resulta en una disminución en la velocidad de la red para los usuarios finales y una degradación en la calidad de los servicios de red.

<sup>5</sup> “QoE es el grado de satisfacción o molestia del usuario de una aplicación o servicio” (ITU-T P.10/G.100, 2017).

Indicator) permiten determinar que la red cumpla con los requisitos de rendimiento establecidos y pueda proporcionar un servicio confiable (ITU Academy, 2019).

El presente trabajo se enfoca en la QoS de los dispositivos de red. En consecuencia, se analiza en profundidad los KPIs específicos para este tipo de QoS. Según Mushtaq & Singh (2017), los indicadores técnicos que determinan la QoS a nivel de los dispositivos de red incluyen el ancho de banda, latencia, tiempo de ida y vuelta, jitter, tasa de pérdida de paquetes y la disponibilidad de la red. A continuación, se detalla cada uno de ellos.

- **Ancho de Banda (AB):** Representa la capacidad del canal de transmisión, es decir, la cantidad máxima de bits de datos que pueden ser transmitidos entre dos puntos en un periodo de tiempo determinado. Alternativamente, se puede entender como la velocidad promedio a la que se transmiten flujos de datos específicos entre dos nodos de la red, además, el AB se mide en bits por segundo (bps). Este KPI desempeña un papel crucial en la Calidad de Servicio (QoS), dado que el crecimiento constante de Internet y la diversificación de servicios pueden convertirlo en un potencial cuello de botella<sup>6</sup> (Huawei, 2023).
- **Latencia:** Es el tiempo que tarda un paquete en viajar desde la fuente hasta el destino, representando un solo sentido del trayecto, como se ilustra en la **Figura 7** (Huawei, 2023). Frecuentemente, puede ser influenciado por la congestión de la red, que ocurre cuando hay una acumulación de paquetes esperando ser transmitidos. Además, este factor puede ser medido mediante la herramienta ping<sup>7</sup> y su unidad métrica son los milisegundos (ms) (Zola, 2024).

---

<sup>6</sup> Cuello de botella es una expresión que se emplea en el ámbito de las redes de comunicación para describir un punto o componente que restringe el flujo de datos, reduciendo así el rendimiento de la red.

<sup>7</sup> Ping es una herramienta que permite evaluar el acceso a la red de un dispositivo específico, medir la velocidad de transferencia de datos entre dos puntos y el tiempo de respuesta en la comunicación entre dispositivos. También es eficaz para solucionar problemas de conectividad (Zola, 2024).

- **Tiempo de viaje de ida y vuelta (RTT):** RTT, abreviado del término inglés Round-Trip Time, se diferencia de la latencia ya que representa el tiempo completo que lleva una solicitud para viajar a través de la red y recibir una respuesta (ver **Figura 7**). Este indicador se mide en milisegundos (AWS, 2023).
- **Jitter:** Al igual que los dos KPIs anteriores, se mide en milisegundos y se define como la variación del retardo en el tiempo entre paquetes consecutivos que forman parte del mismo flujo, como se ilustra en la **Figura 8** (Alarcón, 2003). El jitter puede afectar considerablemente la calidad de la transmisión de datos, particularmente en aplicaciones sensibles al tiempo, como la transmisión de voz y video en tiempo real. Y aunque el buffer<sup>8</sup> de los dispositivos de red puede mitigar los jitters excesivos, también prolonga la latencia (Geeks for Geeks, 2023).
- **Tasa de pérdida de paquetes:** Es la relación entre los paquetes perdidos y el total de paquetes transmitidos. Aunque una pérdida leve de paquetes pueda no impactar en los servicios, es esencial regular este aspecto para prevenir una pérdida significativa de paquetes. QoS se enfoca en controlar la tasa de pérdida de paquetes para garantizar que permanezca dentro de un rango aceptable durante la transmisión de datos en la red (Huawei, 2023).
- **Disponibilidad de red:** Se trata de una medida de la capacidad de una red para hacer frente a las necesidades de conectividad y rendimiento que se le imponen. Este parámetro resulta afectado por varios factores, como fallas en el suministro de energía, daños físicos (desastres naturales), fallos en los equipos y ataques cibernéticos (F5, 2024).

Finalmente, la importancia de QoS es cada vez más alta debido a que la demanda de rendimiento de red se ajusta al creciente número de personas que la emplean. Asimismo, las

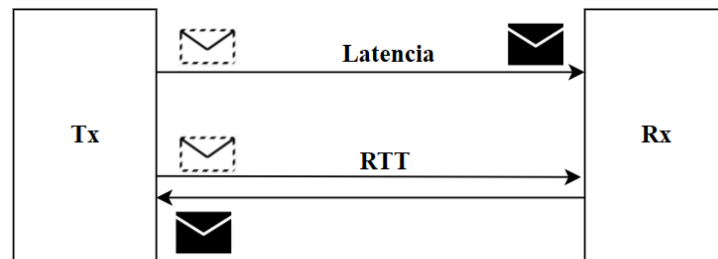
---

<sup>8</sup> Buffer es la memoria temporal que almacena los datos que están a la espera de ser transmitidos.



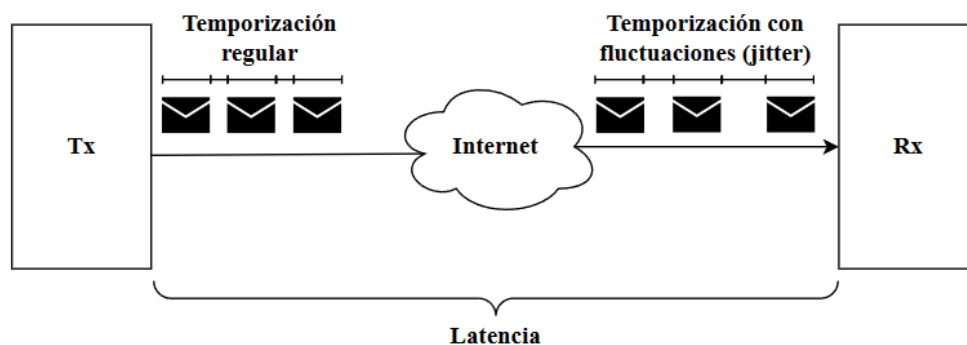
nuevas invenciones tecnológicas necesitan comunicaciones en tiempo real y donde cualquier retraso en la retroalimentación podría causar errores muy costosos, como es el caso de Internet de las Cosas (IoT, Internet of Things) (Fortinet, 2022).

**Figura 7**  
*Diferencia entre Latencia y RTT.*



*Nota.* Tomado y adoptado de Aibin (2024).

**Figura 8**  
*Ejemplo de Jitter.*



*Nota.* Tomado y adoptado de Steed & Fradinho (2009).

### 2.2.1. Modelos de QoS

Los modelos de calidad de servicio son esquemas que determinan el modo de funcionamiento de QoS sobre una red. En las redes IP tradicionales el esquema predeterminado era Best-Effort o “Mejor esfuerzo”, donde todos los paquetes que pasaban por los enrutadores tenían la misma prioridad, de modo que competían por los recursos de red. Sin embargo, dicha característica no es escalable para las aplicaciones sensibles al tiempo que surgieron posteriormente. Frente a esta limitación, el Grupo de Trabajo de Ingeniería de Internet (IETF)

diseñó diversos modelos, donde aquellos comúnmente estudiados y aplicados son Servicios Integrados (IntServ) y Servicios Diferenciados (DiffServ) (Alarcón, 2003).

#### **2.2.1.1. Best-Effort (BE).**

Es el modelo de servicio más simple, puesto a que no proporciona ningún aseguramiento en relación con el ancho de banda, retardo, jitter o pérdida que puedan experimentar los paquetes durante su tránsito (Alarcón, 2003). En este sentido, Best-Effort implica que la red hace su mejor esfuerzo para transmitir cada paquete hacia su destino, pero no ofrece la garantía de que los paquetes lleguen corruptos, duplicados o desordenados (Huawei, 2023).

Dentro de este esquema, las aplicaciones transmiten los datos de manera espontánea y sin limitaciones de cantidad, prescindiendo de solicitar autorización previa ni informar a la red. Es decir que, sin el soporte provisto por protocolos de transporte inteligentes como TCP<sup>9</sup>, este modelo puede desembocar en situaciones caóticas (Alarcón, 2003). Huawei (2023), añade que, BE puede funcionar junto con otros modelos, y es adecuado para servicios que tienen bajos requisitos de retraso y tasa de pérdida de paquetes.

Para finalizar, teniendo en cuenta la simplicidad de Best Effort (BE), caracterizado como un modelo sin Calidad de Servicio (QoS), es importante destacar que el concepto de QoS adquiere mayor relevancia con la estandarización del protocolo IP en el RFC 791. En este documento, la cabecera IP reserva el segundo byte para el campo "Type of Service" (ToS), el cual define la prioridad de los paquetes. Posteriormente, ante el agotamiento de direcciones IPv4, surge IPv6, cuya cabecera presenta diferencias significativas respecto a la versión

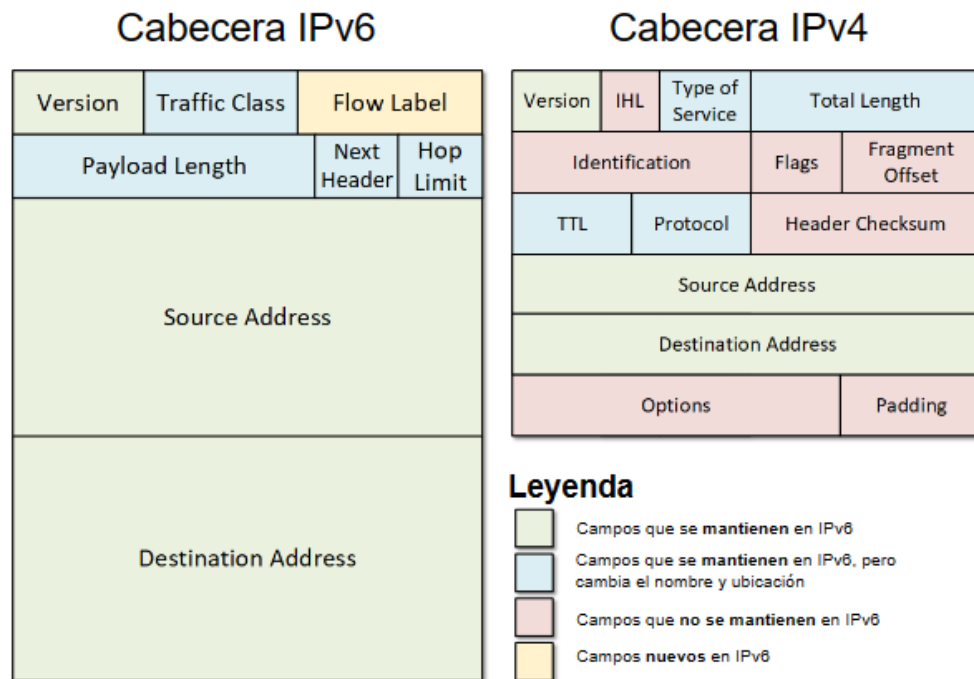
---

<sup>9</sup> TCP (Transmission Control Protocol) es un protocolo esencial en la arquitectura de redes TCP/IP, ubicado en la capa de transporte. Su función principal radica en asegurar la entrega confiable y ordenada de datos desde el emisor hasta el receptor, proporcionando una comunicación estable y sin errores en entornos de red (Fortinet, 2024)(Fortinet, 2024).

anterior. En IPv6, el campo destinado para QoS ahora se denomina "Traffic Class" (TC). Las diferencias entre las cabeceras mencionadas se detallan en la **Figura 9**.

**Figura 9**

*Diferencias entre las cabeceras IPv4 e IPv6.*



*Nota.* Tomado de Network Academy (2023). En la figura se exhibe los campos de las cabeceras IPv4 e IPv6. Cada uno de estos campos desempeña un papel crucial en el proceso de enrutamiento y entrega de datos en redes IP.

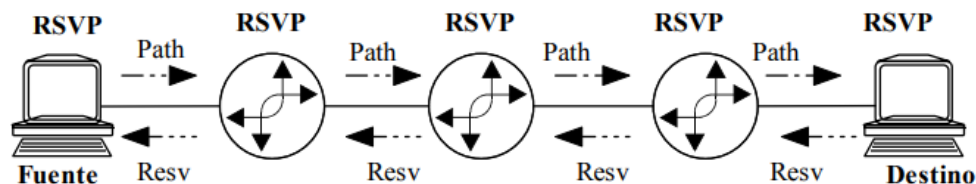
### 2.2.1.2. Modelo de Servicios Integrados (IntServ).

Este modelo se basa en el concepto por flujo, es decir que realiza una reserva de recursos por cada flujo para la diferenciación de servicios (Malik et al., 2015). Utiliza el protocolo de reserva de recursos (RSVP, Resource Reservation Protocol) en donde la aplicación envía una señal a la red solicitando un servicio específico para el cual se garantice un ancho de banda y un retardo máximo aceptable. Posteriormente, la aplicación enviará los datos únicamente después de recibir la confirmación de la red (Alarcón, 2003).

En el funcionamiento del protocolo RSVP intervienen los mensajes Path y Resv, el primero es la solicitud de los recursos de red, mientras que el segundo es la respuesta de la red. Generalmente, las solicitudes RSVP resultan en reservas de recursos en cada nodo a lo largo de la red, como se muestra en la **Figura 10** (Karakus & Durrezi, 2017). Así, el gran número de mensajes necesarios para mantener el estado de las reservas hace que el sistema sea poco escalable (Alarcón, 2003).

**Figura 10**

*Esquema de funcionamiento de IntServ.*



*Nota.* Tomado de Alarcón (2003). La figura muestra el intercambio de mensajes del protocolo RSVP, destacando la participación de los sistemas terminales, tales como computadoras personales y servidores en la comunicación.

### 2.2.1.3. Modelo de Servicios Diferenciados (DiffServ).

Para mitigar el problema de escalabilidad del modelo anterior, surgió el esquema de Servicios Diferenciados (DiffServ), el cual categoriza los paquetes de red en diversas Clases de Servicio (CoS), asignando a cada paquete una prioridad específica que determina las acciones que se llevarán a cabo en función de su clase (Karakus & Durrezi, 2017). La clasificación de paquetes se realiza en el campo ToS para los paquetes IPv4, o en el campo TC para IPv6 y este proceso se denomina “Marcaje”. Además, dicha clasificación se establece únicamente en el dispositivo que proporciona acceso a la red (dispositivo de borde) y, una vez que los paquetes están en la red, el tipo de procesamiento depende del contenido de su encabezado (García, 2007).

En este modelo, la regla de manejo de paquetes se denomina Comportamiento por Salto (PHB, Per Hop Behavior). Así, cada dispositivo de red trata de cierta manera a un conjunto

específico de paquetes que posean el mismo valor de prioridad. PHB se gestiona mediante el DiffServ Code Point (DSCP), el cual permite la clasificación y gestión del tráfico de red, asignando diferentes niveles de servicio a los paquetes de datos en función de su importancia. Según lo establecido por Cisco Systems (2005), la IETF define cuatro PHBs básicos:

- **Default (BE):** Establece el servicio de mejor esfuerzo y se aplica a todo el tráfico que no ha sido identificado o que está marcado con DSCP=0. El valor de los bits DSCP es 000000 (Gonzalez, 2020).
- **Class-Selector (CS):** Es una clasificación que se utiliza para mantener la compatibilidad con los dispositivos que admiten IP Precedence<sup>10</sup> y no DSCP (Gonzalez, 2020). Los valores DSCP se establecen en el formato 'xxx000', donde x puede ser 0 o 1 (Cisco Systems, 2005).
- **Expedited Forwarding (EF):** Ofrece un servicio de red robusto para aplicaciones críticas como VoIP, con baja pérdida de paquetes y latencia. Sin embargo, a pesar de sus ventajas, debe emplearse únicamente en aplicaciones más esenciales debido a las limitaciones durante la congestión. El valor DSCP recomendado para EF es '101110' (Cisco Systems, 2005).
- **Assured Forwarding (AF):** Es un procedimiento que se aplica a los servicios que necesitan un ancho de banda garantizado (Gonzalez, 2023). AF se define por dos parámetros: el Selector de Clase (CS) y la Probabilidad de Descarte (DP), representados como AF(CS; DP). CS utiliza los primeros tres bits, similar a IPP, y su valor puede variar de 1 a 4. DP, por otro lado, utiliza los dos bits siguientes a CS, con el tercer bit siempre en cero, permitiendo valores de 1 a 3 para este campo, donde 1 indica la menor

---

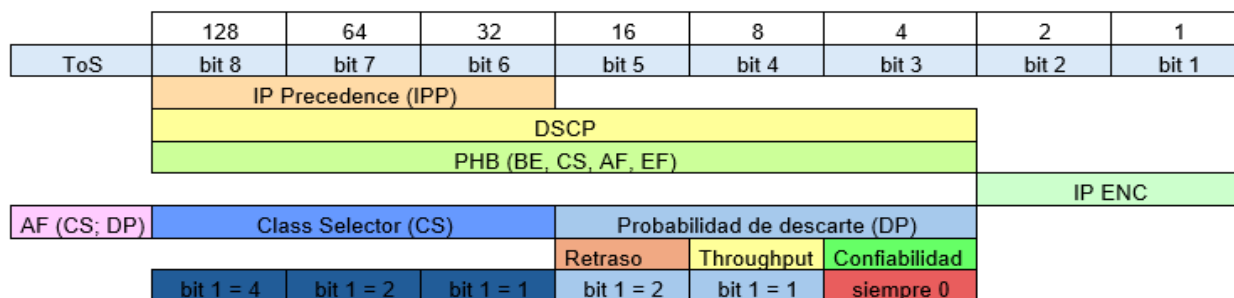
<sup>10</sup> IPP es un modelo de QoS que permite asignar prioridades relativas a los paquetes, pero no distingue entre paquetes con la misma precedencia durante la congestión (Alarcón, 2023).

probabilidad de descarte y 3 la mayor (ver **Figura 11**). En consecuencia, existen cuatro categorías; AF1, AF2, AF3 y AF4 (Moisa, 2021).

El campo ToS/TC ocupa el segundo byte de los paquetes IP. Y los subcampos que comprenden los ocho bits de Tos/TC se encuentran ilustrados en la **Figura 11**, donde: (i) IPP ocupa los primeros tres bits; (ii) DSCP se compone de los tres bits de IPP y los tres siguientes, lo que significa que ocupa un total de seis bits. Y puede representarse numéricamente o mediante caracteres como BE, CS(x), AF(x,y) y EF; (iii) IP ENC (IP Explicit Congestion Notification), sirve para señalar la presencia de congestión en la red y ocupa los dos últimos bits del campo. En este caso, el bit2 indica si se soporta IP ENC, mientras que el bit1 señala la existencia o ausencia de congestión (Moisa, 2021).

**Figura 11**

*Estructura de campo ToS/TC en paquetes IP.*



*Nota. Adoptado de Moisa (2021).*

La asignación del nivel de servicio o prioridad para una aplicación se especifica en el subcampo DSCP, que puede ser tanto numérica como alfanumérica. La **Tabla 2** proporciona una gama de valores asignables a los servicios, ordenados de menor a mayor prioridad, comenzando con servicios de baja prioridad como BE y ascendiendo hacia aquellos de mayor prioridad.

**Tabla 2**  
*Clasificación y Priorización de Tráfico.*

Tráfico o aplicación	IPP = CS	AF	DSCP	ToS	DP
Best Effort	0	0	0	0	
Scavenger <sup>11</sup>	1	CS1	8	32	
Bulk data <sup>12</sup>	1	AF11	10	40	Bajo
	1	AF12	12	48	Medio
	1	AF13	14	56	Alto
Administración de red	2	CS2	16	64	
Datos transaccionales	2	AF21	18	72	Bajo
	2	AF22	20	80	Medio
	2	AF23	22	88	Alto
Señalización de llamadas	3	CS3	24	96	
Misión crítica	3	AF31	26	104	Bajo
Streaming de video	3	AF32	28	112	Medio
	3	AF33	30	120	Alto
	4	CS4	32	128	
Video interactivo	4	AF41	34	136	Bajo
	4	AF42	36	144	Medio
	4	AF43	38	152	Alto
	5	CS5	40	160	
Voz	5	EF	46	184	
Control de red	6	CS6	48	192	
Control de enrutamiento	7	CS7	56	224	

*Nota.* Adoptado de Moisa (2021) e ITU-T Y.1541 (2011).

### 2.2.2. Servicios y sus protocolos

Las aplicaciones de red, que van desde utilitarias hasta recreativas, son la razón de ser de una red de computadoras y han fomentado la integración de Internet en la vida cotidiana. Los protocolos de transporte de dichos servicios son: UDP (User Datagram Protocol), para servicios no confiables y sin conexión, y TCP (Transmission Control Protocol), para servicios confiables y orientados a la conexión. Así, el desarrollo de aplicaciones depende de UDP y TCP, además de protocolos y puertos propios de los servicios. En la **Tabla 3** se presenta un resumen de los

<sup>11</sup> Scavenger es la denominación que reciben los servicios de baja prioridad.

<sup>12</sup> Bulk Data son grandes cantidades de datos que no requieren entrega inmediata.

servicios multimedia más comunes, incluyendo sus características distintivas, protocolos y puertos de funcionamiento (Kurose & Ross, 2017).

**Tabla 3**  
*Características de los principales servicios multimedia.*

Servicio	Características	Protocolos	Puertos
Voz sobre IP*	Sensible al tiempo y pérdida de paquetes	SIP	5060/UDP-TCP
		RTP	10000-65000/UDP
Video conferencia	Sensible al ancho de banda, retardo y jitter	H.323	1720/TCP
Video bajo demanda*	Sensible al ancho de banda y jitter	HTTP	80/TCP
		RTSP	554/TCP
Audio bajo demanda	Sensible al jitter	HTTP	80/TCP
		HLS	443/TCP
Correo electrónico	Sensible a la pérdida de paquetes	SMTP	25/TCP
		IMAP	143/TCP
		POP3	110/TCP
Transferencia de archivos*	Sensible al ancho de banda y pérdida de paquetes	FTP	21/TCP
		SFTP	22/TCP
Acceso web	Sensible al jitter	HTTP	80/TCP
		HTTPS	443/TCP

*Nota.* Adoptado de Alkahtani et al. (2003) y Buñay et al. (2019). Los servicios marcados con asterisco (\*) son los que se encuentran presentes en el desarrollo de este trabajo.

### 2.2.3. Normativas de QoS

Las redes contemporáneas y emergentes requieren ofrecer una Calidad de Servicio diferenciada y segura para la creciente gama de aplicaciones. Para implementar soluciones de QoS de extremo a extremo, los proveedores de redes IP deben llegar a un consenso sobre un conjunto uniforme de parámetros de rendimiento para la transferencia de paquetes IP y objetivos de QoS. Por esta razón, se han establecido regulaciones internacionales como las Recomendaciones ITU-T E.800, ITU-T Y.1540 e ITU-T Y.1541, mismas que proporcionan directrices y estándares que ayudan a los proveedores de redes a garantizar una QoS óptima y consistente en las redes, permitiendo así una mayor eficiencia en la gestión de la red (Seitz, 2003). Además, es importante destacar que estas recomendaciones están accesibles al público, lo que representa una ventaja significativa.



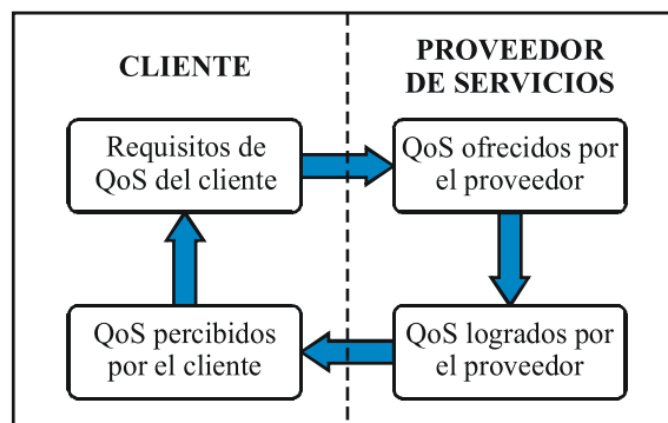
### 2.2.3.1. ITU-T E.800.

Es un documento esencial que proporciona definiciones de términos relacionados con el estudio y gestión de QoS en el ámbito de telecomunicaciones. La terminología abarca aspectos técnicos y no técnicos, lo que la hace relevante para una amplia gama de las partes interesadas en el sector, incluyendo proveedores de servicios, fabricantes de equipos y usuarios finales. La importancia de definir una terminología común es crucial para evitar malentendidos entre los usuarios de las normas debido a términos y definiciones contradictorios, y para facilitar la coordinación entre los diferentes grupos involucrados en la creación de normas de telecomunicaciones (ITU-T E.800, 2008).

Los conceptos presentados por esta norma se vinculan con los cuatro polos de QoS, donde se presentan las expectativas y requerimientos del cliente, así como el nivel de servicio ofrecida y lograda por el proveedor, tal como lo ilustra la **Figura 12**.

**Figura 12**

*Los cuatro polos de la Calidad de Servicio.*



*Nota. Tomado de ITU-T E.800 (2008).*

### 2.2.3.2.ITU-T Y.1540.

Esta recomendación define los parámetros que pueden emplearse para evaluar la calidad de funcionamiento en la transferencia unidireccional de paquetes en redes IP en función de velocidad, exactitud y seguridad de funcionamiento. Dichos parámetros se aplican a un servicio IP de extremo a extremo, punto a punto, y a tramos de la red que proporcionan, o contribuyen a la prestación de este servicio (ITU-T Y.1540, 2019).

Para establecer los parámetros de medición, la recomendación establece un modelo de calidad de funcionamiento de un servicio IP, en donde define los conceptos más relevantes dentro de una red con los que se puede representar a un servicio IP de extremo a extremo. Entre los conceptos se definen; los componentes de red, los enlaces y las secciones de red, los puntos de medición y secciones sensibles, los eventos de referencia de en la transferencia de paquetes y los resultados en la transferencia de paquetes.

- **Componentes de red:** Los componentes de red representan los equipos y conexiones existentes en la red, además, para cada uno de ellos se establecen las siglas correspondientes que permitirá identificarlos en la topología de red. Estas siglas se presentan en la **Tabla 4**.
- **Enlace central y secciones de red:** Un enlace central (EL) es aquel que conecta un computador principal de origen o destino con un encaminador, así, un enlace central puede denominarse también como un enlace de acceso. Por otro lado, una sección de red (NS) es cualquier conjunto de componentes interconectados por enlaces, y que en conjunto proporcionan un servicio IP entre un SRC y un DST. Además, dichos componentes poseen el mismo identificador de red en sus direcciones IP.
- **Puntos de medición y secciones medibles:** Un punto de medición (MP) es la frontera entre un computador principal y un enlace adyacente en donde se puede observar y medir eventos referentes a la calidad de funcionamiento de la red. La

recomendación define tres tipos de secciones medibles; (i) Sección básica, que puede ser un EL, una NS, un SRC o un DST; (ii) Red IP de extremo a extremo; (iii) Conjunto de secciones de red (NSE), el cual representa conexiones entre diversos NS. Es decir que, una sección es medible si está limitada por un conjunto de puntos de medición.

- **Eventos de referencia en la transferencia de paquetes IP (IPRE):** Un evento de referencia ocurre cuando un paquete IP cruza un punto de medición, se verifica que no esté dañado y que las direcciones de origen y destino sean correctas.
- **Resultados en la transferencia de paquetes IP:** Considerando los eventos de referencia en la transferencia de paquetes IP, es posible definir varios resultados posibles, tales como; transferencia satisfactoria, errores en la transferencia, paquetes perdidos y paquetes espurios<sup>13</sup>. Además, dichos resultados se establecen en función de los MP de ingreso y egreso correspondientes, como se evidencia en la **Figura 13**.

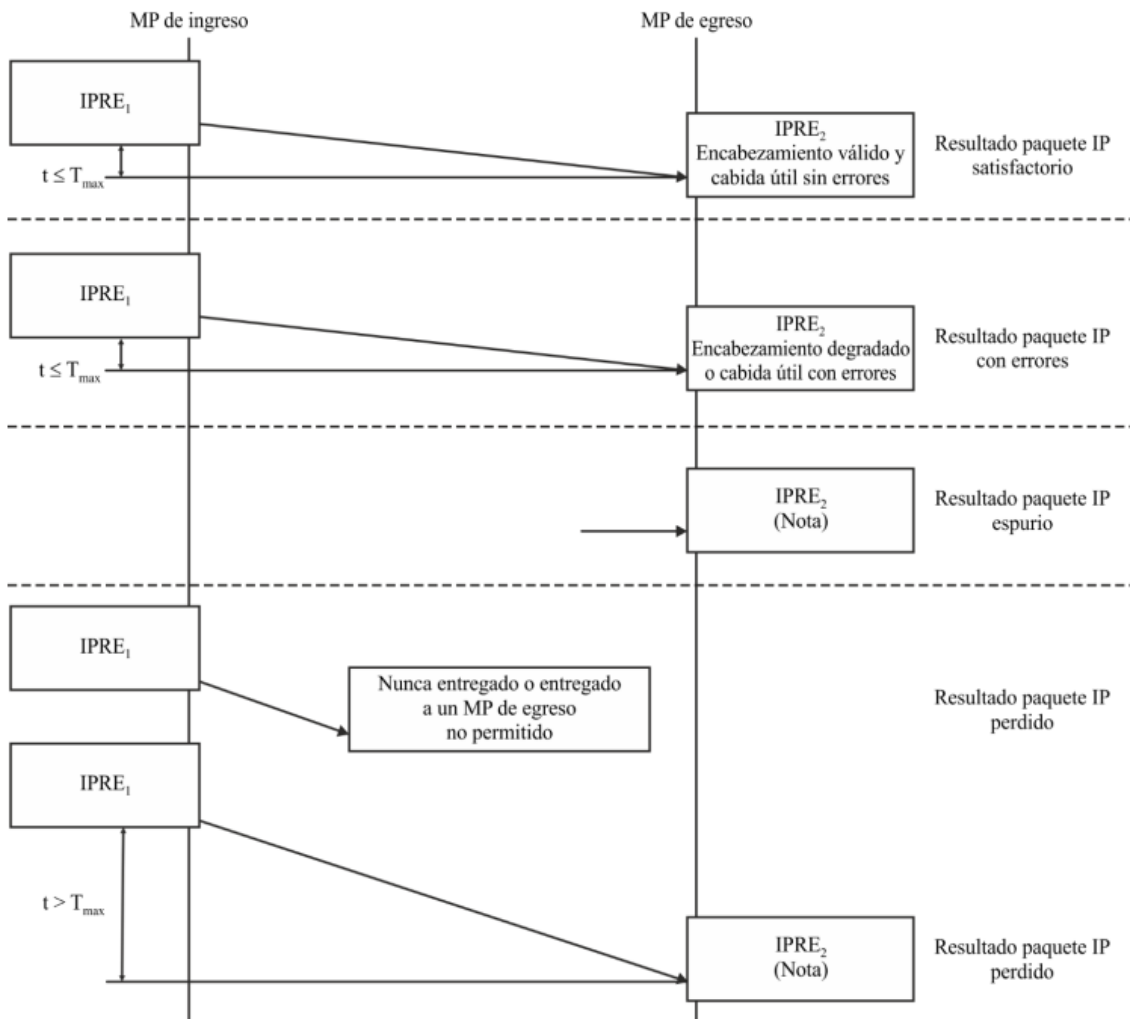
**Tabla 4**

*Siglas y representación de componentes de red definidos en ITU-T Y.1540.*

<b>Componente</b>	<b>Siglas</b>	<b>Definición</b>
Encaminador	R	Permite reenviar los paquetes IP en función del contenido en el campo de dirección IP de destino. Este componente se representa con un círculo.
Enlace	L	Conexión punto a punto empleada para el transporte de paquetes IP. La representación de un enlace es mediante una línea simple.
Computador principal de origen	SRC	Es en donde se originan los paquetes IP. Es representado con un triángulo.
Computador principal de destino	DST	Es hacia donde llegan los paquetes IP. Y al igual que SRC, es representado con un triángulo.

<sup>13</sup> Se denominan paquetes espurios a aquellos recibidos en el destino, pero que no han sido transmitidos desde el transmisor, también se conocen como paquetes IP falsos ITU-T Y.1540 (2019).

**Figura 13**  
Resultados posibles en la transferencia de paquetes IP.



*Nota.* Tomado de ITU-T Y.1540 (2019). La figura ilustra la definición de los cuatro resultados posibles en la transferencia de paquetes IP.

La mayoría de los parámetros de calidad de funcionamiento en la transferencia de paquetes IP se definen en base a una población de interés, dicha población es un flujo de paquetes IP que se envían desde un SCR hacia un DTS y que cruzan por un MP específico (ITU-T Y.1540, 2019). La recomendación establece doce parámetros, los cuales se detallan a continuación.

- 1. Retardo de transferencia de paquetes IP (IPTD):** También conocido como latencia, es el indicador que se define en función de  $t_2 - t_1$ , siendo  $t_2$  el tiempo de llegada del paquete al destino y  $t_1$  el tiempo de salida desde la fuente, donde  $t_2 > t_1$ . De acuerdo

con la ITU-T Y.1540 (2019), de este parámetro se derivan otros más, de los cuales se destaca el siguiente:

- a. Variación del retardo de paquetes IP (IPDV):** Es la diferencia del retardo de un paquete IP y un retardo de referencia entre los mismo fuente y destino de la comunicación. En muchos textos se denomina Jitter.
- 2. Tasa de errores en los paquetes IP (IPER):** Es la relación entre la cantidad de paquetes erróneos y el total de paquetes transmitidos.
- 3. Tasa de pérdida de paquetes IP (IPLR):** Es la relación entre el total de paquetes perdidos y el total de paquetes transmitidos.
- 4. Tasa de paquetes IP espurios (SIPR):** Es el número total de paquetes espurios observados durante un intervalo de tiempo específico y dividido para la duración del intervalo, así, se expresa como una tasa de tiempo, ya que los mecanismos que causan estos paquetes tienen poco que ver con la cantidad de paquetes transmitidos.
- 5. Tasa de reordenación de paquetes IP (IPRR):** Es la relación entre el total de paquetes reordenados y el total de paquetes IP transmitidos de forma exitosa.
- 6. Tasa de bloques con pérdidas severas de paquetes IP (IPSLBR):** Es la relación entre la cantidad de bloques con pérdidas y el número total de bloques dentro de una población de interés.
- 7. Tasa de duplicación de paquetes IP (IPDR):** Es la relación entre la cantidad de paquetes duplicados y la cantidad de paquetes transmitidos satisfactoriamente, menos la cantidad de paquetes duplicados.
- 8. Tasa de paquetes IP replicados (RIPR):** Es la relación entre el total de paquetes IP replicados y el total de paquetes transmitidos de forma satisfactoria, menos la cantidad de paquetes duplicados.

- 9. Parámetros de reparación de trenes:** Permiten determinar la cantidad de paquetes o bloques degradados en la transmisión de una población de interés. Para los dos casos, el cálculo es una relación entre los resultados de los paquetes/bloques IP degradados y el total de intervalos no degradados dentro de una población de interés.
- 10. Parámetros de capacidad:** Para una sección de red básica (NS) y una determinada población de interés, la capacidad se expresa con la **Ec. 1**. En dicha ecuación  $n_0$  representa el total de bits transmitidos de forma exitosa dentro de un intervalo de tiempo específico  $\Delta t$ .

$$C(t, \Delta t) = \frac{n_0(t, \Delta t)}{\Delta t} \quad (\text{Ec. 1})$$

- 11. Parámetros relacionados con el flujo:** Estos parámetros caracterizan la capacidad del caudal de la red en función de una tasa de transferencia estable. De este modo, estos parámetros son la relación entre la cantidad de paquetes IP transportados de manera satisfactoria con la cantidad de paquetes IP totales.
- 12. Disponibilidad e indisponibilidad de red:** Esta se determina a través de la tasa de pérdida de paquetes (IPLR). Así, para un flujo específico, su disponibilidad durante un período de observación se define si se cumple la expresión de **Ec. 2**, donde  $c_1$  es un valor de referencia de 0,20. De lo contrario, se considerará no disponible. Esto se debe a que muchas aplicaciones de redes IP dejan de funcionar cuando la tasa de pérdidas es mayor a 0,20 (ITU-T Y.1540, 2019).

$$IPLR < c_1 \quad (\text{Ec. 2})$$

También, la disponibilidad e indisponibilidad de servicios IP se relacionan mediante la **Ec. 3**, donde PIA son las siglas de *porcentaje de disponibilidad*, mientras que PUI son las siglas de *porcentaje de indisponibilidad*. Además, para este cálculo se considera la función de disponibilidad de servicio IP (Ec. 2).

$$PUI = 100 - PIA \quad (\text{Ec. 3})$$

Por otro lado, el proceso relacionado con la evaluación de calidad de funcionamiento se presenta en dos pasos esenciales; (i) Definición de las interfaces en las que se aplicarán los parámetros y eventos específicos para ser medidos, cronometrados o comparados con evento de referencia (RE). Este punto considera la población de interés, así como las herramientas que se empleará para medir los eventos de referencia. (ii) Establecimiento de un conjunto de KPIs que caractericen el desempeño de la red (ITU-T Y.1540, 2019).

El alcance de esta recomendación se representa en la **Figura 14**, donde es posible evidenciar que, tomando como punto de partida el modelo o estructura de la red, se definen los componentes de red, eventos de referencia e interfaces de medición en donde se evaluará su rendimiento. Para dicho escenario se define un conjunto de parámetros de red que caracterizan el funcionamiento de la misma, dichos parámetros se evalúan en base a los criterios de velocidad, exactitud y seguridad de funcionamiento, así mismo, la recomendación establece parámetros de medición para determinar la disponibilidad de la red.

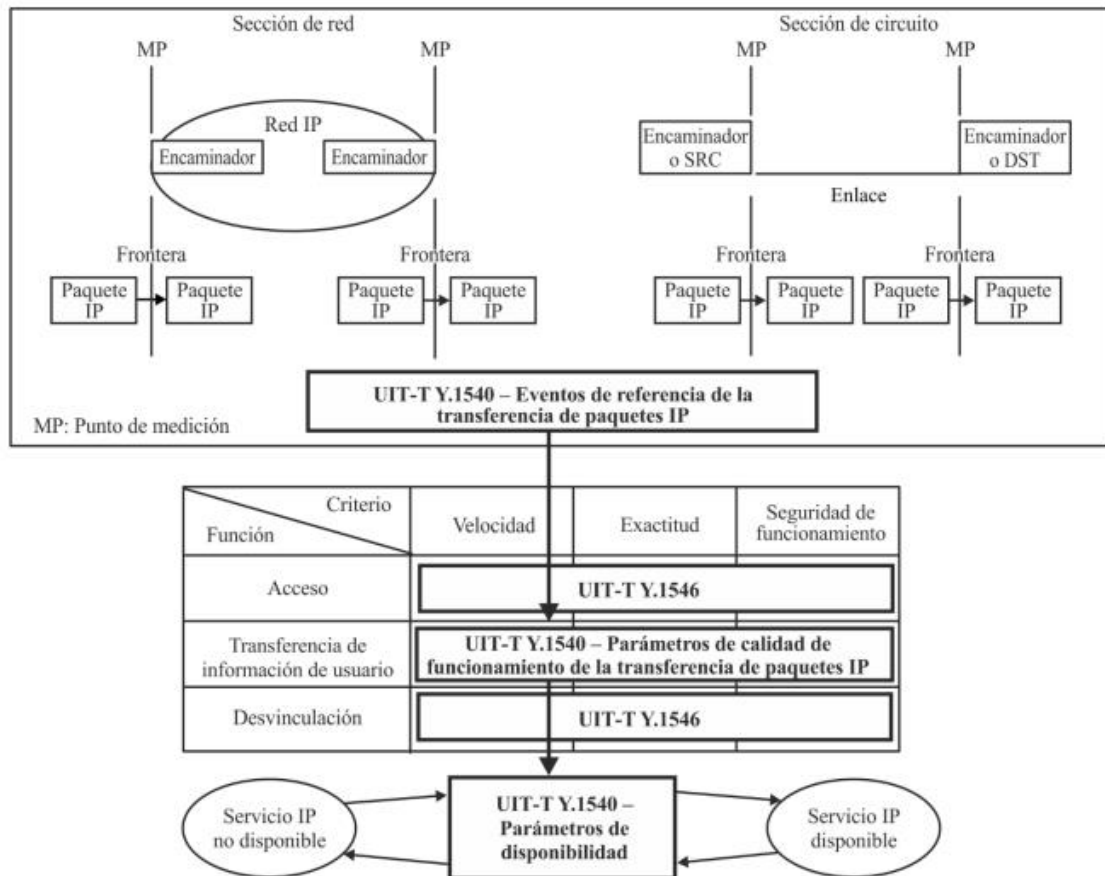
Finalmente, de acuerdo con Seitz (2003), esta normativa puede ser empleada por tres entes principales; proveedores, fabricantes y usuarios. Los proveedores la emplean para realizar la planificación, desarrollo y evaluación de redes. Los fabricantes, por su parte, la usan para el desarrollo de equipos y la implementación de estrategias de marketing, conforme a las especificaciones requeridas por los proveedores. Mientras que los usuarios la pueden emplear para evaluar el rendimiento que ofrecen las redes IP en los terminales, sin embargo, este proceso está estrechamente vinculado con el Acuerdo de Nivel de Servicio<sup>14</sup> (SLA) determinado con el proveedor.

---

<sup>14</sup> SLA es un contrato entre un cliente y un proveedor de servicios. Describe el servicio a proporcionar, el rendimiento esperado, los métodos de medición y evaluación, y las consecuencias de no alcanzar los niveles de rendimiento establecidos (Goodwin, 2023).

**Figura 14**

Alcance de la recomendación ITU-T Y.1540.



### 2.2.3.3. ITU-T Y.1541.

Establece valores numéricos que deben alcanzarse para los KPIs definidos en la Recomendación ITU-T Y.1540. Dichos valores se agrupan en varias clases de QoS. Cada clase de QoS establece metas específicas para lograr parámetros de rendimiento óptimos en las redes IP, como se muestra en la **Tabla 5**.

Finalmente, esta recomendación considera el crecimiento de las redes IP futuras, es por ello que permite la aplicación de QoS basado en Servicios Diferenciados definido por la IETF. Así, la relación entre las dos normativas, se muestran en la **Tabla 6**.



**Tabla 5**

*Definición de clases de QoS y objetivos de rendimiento de la red.*

Parámetro	Objetivo de rendimiento	Clases de QoS					
		Clase 0	Clase 1	Clase 2	Clase 3	Clase 4	Clase 5
IPTD	Límite superior de la media del IPTD.	100 ms	400 ms	100 ms	400 ms	1 s	U
IPDV	Límite superior en el cuantil 1 – 10 <sup>3</sup> del IPTD menos el IPTD mínimo.	50 ms	50 ms	U	U	U	U
IPLR	Límite superior de la IPLR	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	U
IPER	Límite superior	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	U

*Nota.* Tomado de ITU-T 1541. La letra U presente en la tabla significa no especificado.

**Tabla 6**

*Asociación de clases de QoS de UIT-T Y.1541 con PHB de DiffServ.*

PHB DiffServ Asociado	Clases de QoS	Observaciones
Default (BE)	Clase 5	Un servicio, cuando se opera en una red con poca carga, puede alcanzar un buen nivel de QoS.
AF	Clase 2, 3 y 4	El IPLR solo se aplica a los paquetes IP en los niveles de prioridad más altos de cada clase AF. El IPTD se aplica a todos los paquetes.
EF	Clase 0 y 1	

*Nota.* Adoptado del Apéndice VI de la Recomendación ITU-T Y.1541.

### 2.3. Redes inalámbricas definidas por software (SDWN)

Principalmente el concepto de Redes Definidas por Software (SDN, Software Defined Networking) surgió como una solución a las restricciones presentes en las estructuras de red convencionales. Donde sus principales ventajas son la visión global y centralizada de la red, la programabilidad y la separación del plano de datos y el plano de control (Karakus & Durresi, 2017). Posteriormente, con el creciente despliegue y diversificación de la tecnología

inalámbrica, la gestión de estas redes se volvió un desafío considerable. En respuesta a esto, se contempló la incorporación de las redes inalámbricas en el paradigma SDN, lo que ha dado lugar a las Redes Inalámbricas Definidas por Software (SDWN) (Malik et al., 2015).

El enfoque de SDWN abarca las tecnologías de redes móviles, inalámbricas regidas por el estándar 802.11 (WLAN) y las redes de sensores. Cada una de estas tecnologías presenta diversos esquemas o topologías de configuración que pueden ser adoptados desde los esquemas convencionales al paradigma de SDWN, o bien, nuevos esquemas propuestos por la comunidad para aprovechar todas las ventajas de SDN. A continuación, se describen las topologías más comunes utilizadas en escenarios SDWLAN: LVAP y One Big AP (Dezfouli et al., 2018).

- **Punto de Acceso Virtual Ligerito (LVAP):** Este esquema ofrece una interfaz de alto nivel para controlar el estado de un cliente inalámbrico. La implementación de esta interfaz se encarga de aspectos como la asociación, la autenticación y la gestión de recursos. Así, cuando un cliente inalámbrico intenta conectarse a la red, se activa la creación de un nuevo LVAP. En consecuencia, cada AP alojará tantos LVAP como el número de clientes que actualmente estén comunicándose con él. En definitiva, el LVAP se considera como un AP Virtual con su propio BSSID<sup>15</sup> (Riggio et al., 2014).
- **Ilusión de un Gran Punto de Acceso (One Big AP):** Es una solución empresarial para redes WLAN convencionales que crea la impresión de que un cliente compatible con 802.11 está conectado a un punto de acceso con una cobertura muy amplia, aunque en realidad, el punto de acceso que maneja los paquetes ha cambiado. La implementación de este esquema en un entorno SDN ofrece ventajas: la asociación de AP controlada por la red mejora el rendimiento, mientras que las transferencias transparentes de AP

---

<sup>15</sup> BSSID son las siglas de Basic Service Set Identifier. Se utiliza para identificar de manera única a un punto de acceso inalámbrico en una red (generalmente es la dirección MAC del equipo). Esta identificación permite la conexión y comunicación de dispositivos con el punto de acceso específico.

reducen la latencia. En conjunto, mejora la eficiencia y la gestión de las redes inalámbricas (Dong et al., 2014).

### 2.3.1. *Arquitectura SDWN*

SDN proporciona una arquitectura que permite una gestión de infraestructura de comunicaciones más eficiente y centrada en la programabilidad, con el propósito de optimizar el rendimiento, el monitoreo y la escalabilidad. Así, la Open Networking Foundation<sup>16</sup> (ONF) define la arquitectura SDN en tres planos principales; Administración, Control y Datos o Infraestructura, tal como lo ilustra la **Figura 15** (Karakus & Durresi, 2017).

- **Plano de administración:** También conocido como capa de aplicación, está compuesto por una o varias aplicaciones, y permite que cada una de ellas tenga control exclusivo sobre un conjunto de recursos proporcionados por uno o más controladores SDN (ONF, 2014). Estas aplicaciones se comunican con el controlador mediante la interfaz Northbound, que es una API<sup>17</sup> abierta diseñada para la programación de aplicaciones (Karakus & Durresi, 2017).
- **Plano de control:** La capa de control incluye uno o más controladores SDN implementados en software. Su función es similar a la de un instructor, ya que toma todas las decisiones de conmutación basándose en la topología y facilitan la comunicación entre los dispositivos de red con las aplicaciones. Por esta razón, se le considera el núcleo de la red definida por software (Karakus & Durresi, 2017).
- **Plano de datos:** En la capa de infraestructura se encuentran los componentes de red programables, mismos que son responsables de la conmutación de los datos que fluyen a través de la red. Estos componentes incluyen; routers, switches físicos/virtuales, y

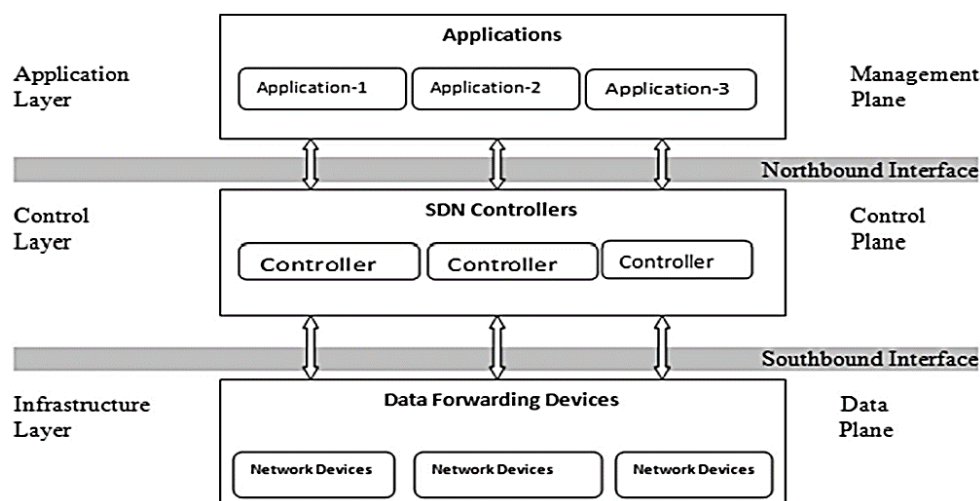
---

<sup>16</sup> ONF es una organización sin fines de lucro que fomenta y democratiza la innovación en redes programables definidas por software (ONF, 2024).

<sup>17</sup> API (del inglés Application Programming Interface), hace referencia a un conjunto de reglas o protocolos que facilitan la comunicación entre diferentes aplicaciones de software, permitiendo el intercambio de datos, atributos y funciones (IBM, 2023).

puntos de acceso (Gonzalez, 2020). Estos dispositivos pueden ser accedidos a través de las Interfaces Southbound, siendo OpenFlow el estándar predominante en dichas interfaces (Karakus & Durresi, 2017).

**Figura 15**  
*Arquitectura SDN.*



*Nota.* Tomado de Keshari et al. (2021).

### 2.3.2. Componentes de red SDWN

Siguiendo la arquitectura de las redes SDN, los componentes se integran de manera eficiente con las funcionalidades de cada capa. De este modo, los componentes clave de esta estructura incluyen; el controlador SDN, los dispositivos programables presentes en la capa de infraestructura, y las interfaces que facilitan la comunicación del controlador con las capas de aplicación e infraestructura.

#### 2.3.2.1. Controlador SDN.

Existen numerosos controladores SDN tanto de código abierto como comerciales disponibles, y las características de cada plataforma de control están diseñadas para satisfacer las necesidades específicas de distintas aplicaciones. En general, los controladores se dividen en dos categorías principales: controladores distribuidos y controladores centralizados. El primero gestiona la lógica del plano de control desde una ubicación única, suele enfrentar

dificultades de escalabilidad debido a las limitaciones de capacidad. Mientras que el segundo no experimenta problemas de escalabilidad y ofrece un rendimiento superior en momentos con alta carga de tráfico (Keshari et al., 2021). La **Tabla 7** proporciona una visión detallada de los principales controladores y sus características más destacadas.

**Tabla 7**

*Comparación de controladores SDN.*

<b>Controlador</b>	<b>Lenguaje</b>	<b>Código Abierto</b>	<b>Soporte de OpenFlow</b>	<b>Distribuido</b>	<b>Rendimiento</b>
ONOS	Java	Sí	Sí	Sí	Alto
Ryu	Python	Sí	Sí	Sí	Medio
POX	Python	Sí	Solo versión 1.0	No	Bajo
OpenDaylight	Java	Sí	Sí	Sí	Medio
Onix	C, Python	Sí	Sí	Sí	Medio
Beacon	Java	Sí	Sí	No	Medio
FloodLight	Java	Sí	Sí	No	Medio

*Nota. Tomado y adoptado de Karakus & Durrezi (2017).*

### **2.3.2.2. Interfaces SDN.**

En las redes SDN, el controlador se comunica con el plano de datos a través de la interfaz Southbound y con las aplicaciones a través de la interfaz Northbound. Las APIs Northbound, comúnmente basadas en RESTful, son esenciales para la interacción de las aplicaciones con la red, manteniendo la simplicidad y evitando la complejidad de las APIs Southbound.

Las interfaces Southbound son APIs que pueden ser tanto de código abierto como propietarias. Entre estas, destaca el protocolo OpenFlow, definido por la ONF. Este protocolo es reconocido como la primera interfaz southbound de su tipo y es ampliamente reconocido en el campo. OpenFlow permite la adición y eliminación de entradas en la tabla de flujo interna de conmutadores y enrutadores, lo que facilita una respuesta más eficiente de la red ante las demandas de tráfico en tiempo real. Este protocolo es de código abierto y hasta la fecha, la versión más reciente es la 1.5 (SDxCentral, 2024).

Por otro lado, las interfaces Northbound suelen ser del tipo API RESTful, las cuales se adhieren a los principios de la arquitectura de Transferencia de Estado Representacional (REST), lo que las hace escalables, flexibles y fáciles de mantener. Emplean métodos HTTP (GET, POST, PUT, PATCH, DELETE) para recuperar o modificar datos. Así, se pueden utilizar para facilitar la automatización de la red y alinearse con las necesidades de diferentes aplicaciones a través de la programabilidad de la red SDN (Cronnor, 2023).

### **2.3.2.3. Puntos de acceso.**

En los escenarios de SDWLAN, los puntos de acceso son dispositivos programables, donde el Open Virtual Switch (OVS) juega un papel crucial. OVS es un proyecto de código abierto que facilita a los hipervisores la virtualización de la capa de red. Uno de sus objetivos primordiales es el énfasis en el alto rendimiento, logrado a través de la utilización de los componentes del kernel de Linux (Fouaz, 2021).

### **2.3.3. Gestión de calidad de servicio en SDWN**

Las redes IEEE 802.11 habilitadas para SDN ofrecen la ventaja de una rápida implementación de QoS. En un entorno SDN, la asignación de ancho de banda, la limitación de velocidad y la configuración del tráfico pueden realizarse eficientemente a nivel de red mediante el uso de APIs automatizadas de QoS a través del controlador de red. Este proceso se beneficia de mediciones en tiempo real que orientan la aplicación de políticas para optimizar los mecanismos de QoS. Dichas políticas, establecidas en el controlador, contribuyen a minimizar el uso innecesario de recursos (Malik et al., 2015).

Las arquitecturas de OpenFlow y SDN pueden ser extremadamente beneficiosas para la automatización del control escalable de QoS en la red, basándose en descripciones de alto nivel de las necesidades de las aplicaciones y servicios. El protocolo OpenFlow, en sus diversas versiones, incorpora una gama de funcionalidades vinculadas a QoS. En la versión 1.0, se

presenta “enqueue”, una acción opcional para administrar colas en los puertos. La versión 1.1 mejora el soporte para VLANs y MPLS. Con OpenFlow 1.2, se añade la posibilidad de consultar todas las colas en un switch y se definen propiedades adicionales de cola. OpenFlow 1.3 introduce tablas de medición para regular la tasa de tráfico, mientras que la versión 1.4 establece un marco para el monitoreo de flujos. Por último, en OpenFlow 1.5, la instrucción de medición se reemplaza por una acción de medición (Malik et al., 2015).

Finalmente, Keshari et al. (2021) señalan que, a pesar del crecimiento en la investigación de SDN y las mejoras en QoS, todavía es necesario profundizar en los estudios en este campo. Los retos de investigación para optimizar la QoS en las redes SDN son los siguientes:

- **Interacción entre diferentes Controladores y Conmutadores:** Cuando se sitúan múltiples switches y controladores en el dominio de redes programables, pueden implementarse diversas configuraciones de comunicación en los dispositivos. Donde diversos proveedores pueden implementar políticas de comunicación distintas en los dispositivos, lo que puede generar conflictos de comunicación en la red (Keshari et al., 2021).
- **Protocolo estándar para el Plano de Control al Plano de Administración:** SDN utiliza un protocolo de código abierto para comunicar el plano de control con el plano de administración. Para los controladores múltiples de gran escala, es muy difícil escalar las interfaces de administración. Por lo tanto, un protocolo estándar en dicha comunicación podría reducir la latencia de la comunicación y ser útil para el equilibrio de carga (Keshari et al., 2021).

### 3. CAPÍTULO III: Diseño de red e implementación de QoS

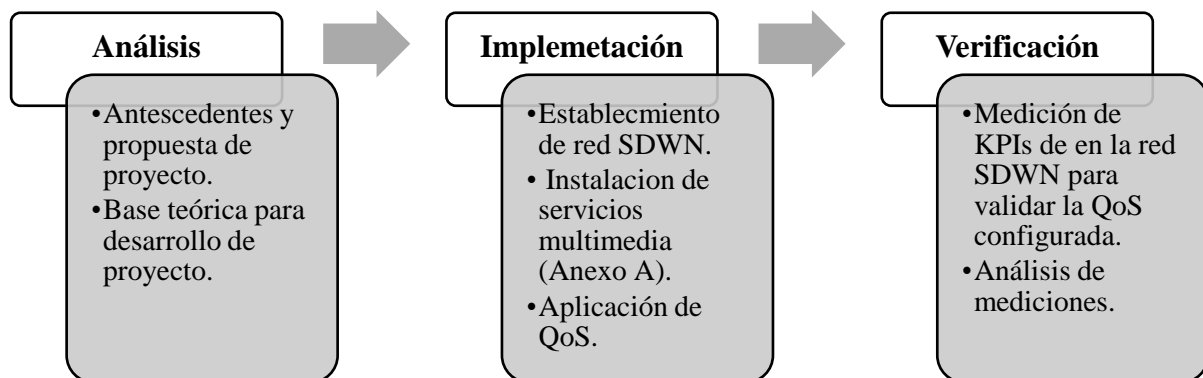
Este capítulo aborda la implementación del prototipo de red SDWN y la configuración de QoS en dicho esquema. El prototipo de red se rige en función del trabajo presentado por Moncayo (2023). Por otro lado, la QoS se considera una aplicación de Servicios Diferenciados, siguiendo las recomendaciones de la ITU-T Y.1541. Para lograr esta implementación, se dividen las fases de desarrollo en análisis, implementación y verificación.

#### 3.1. Fases de desarrollo

Las fases describen un proceso jerárquico que permite la consolidación del proyecto, y se encuentran distribuidas a lo largo de los cuatro capítulos de este documento. La fase de análisis se desarrolla en los capítulos I y II, la implementación se aborda en el capítulo III, y la verificación se lleva a cabo mediante el método analítico-explicativo, detallado en el capítulo IV. En la **Figura 16** se presentan los procedimientos seguidos en cada fase.

**Figura 16**

*Metodología para desarrollo de proyecto.*



#### 3.2. Establecimiento de prototipo SDWN

El prototipo SDWN toma como base el trabajo presentado por Moncayo (2023), el cual se fundamenta en la metodología PPDIOO, establecida por Cisco. Dicha metodología comprende seis fases; Preparar, Planificar, Diseñar, Implementar, Operar y Optimizar. Estas fases definen el ciclo de vida del diseño de redes, abarcando desde la etapa inicial de preparación y una



planificación meticulosa, hasta la implementación efectiva, la operación ininterrumpida y la optimización para potenciar tanto el rendimiento como la eficiencia. En este contexto, las características presentadas por dicho prototipo abarcan tres aspectos específicos; red SDWN, software y hardware, los cuales se detallan en la **Tabla 8**. Para adaptar este prototipo a las necesidades específicas de este trabajo, se han realizado las siguientes adaptaciones:

- Se ha ampliado el número de nodos en la red, considerando el uso de cuatro nodos.
- Se ha seleccionado el firmware OpenWrt y el módulo OVS en sus versiones más actualizadas, para la integración con las últimas funcionalidades disponibles.

**Tabla 8**

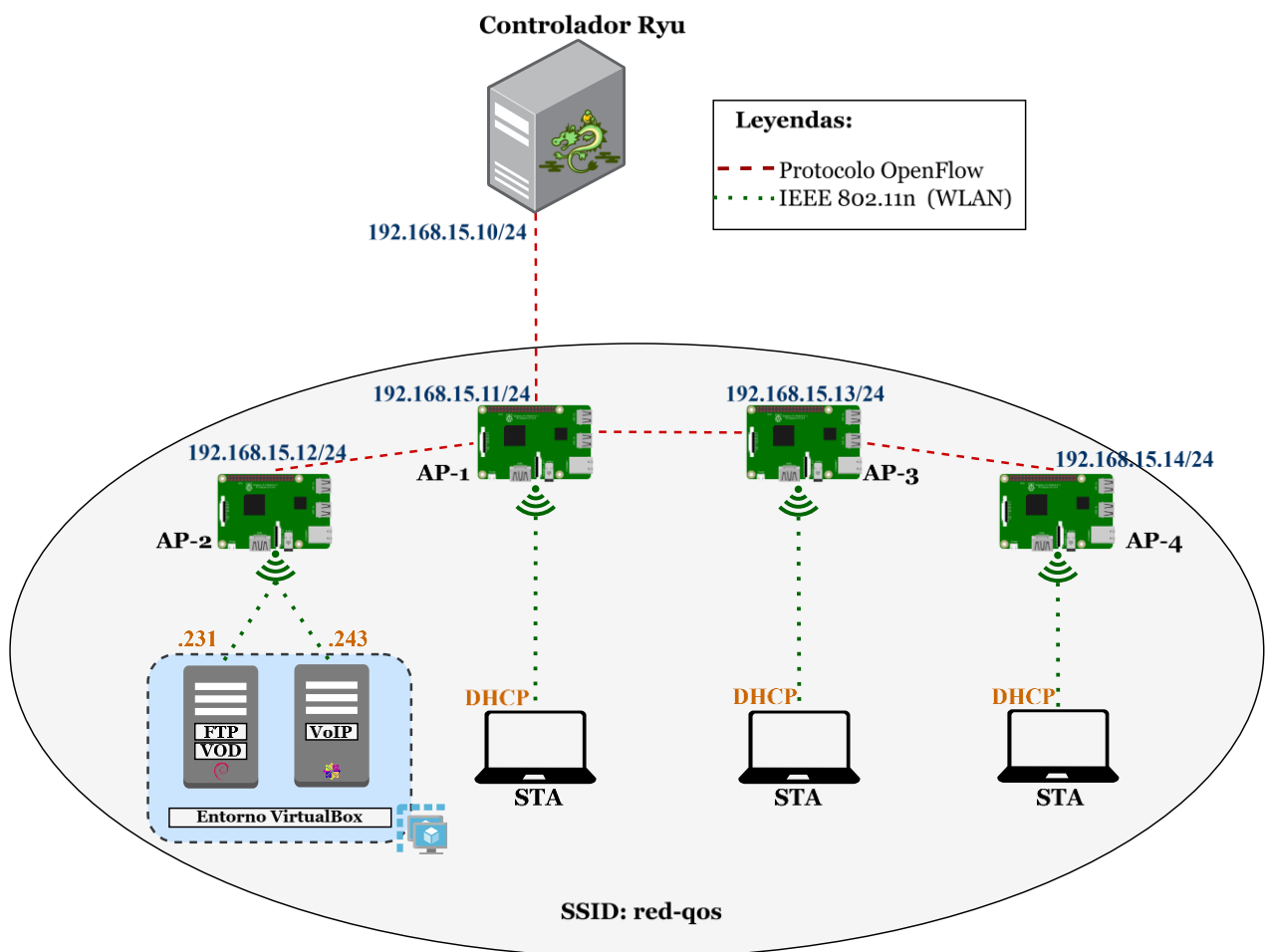
*Características de prototipo SDWN definido por Moncayo (2023).*

<b>Característica</b>	<b>Especificación</b>
<b>Características de red SDWN</b>	
Número de nodos	Dos nodos
Topología de red	One Big AP
Protocolo inalámbrico	IEEE 802.11n
Frecuencia de operación	2.4 GHz
Canales de operación	Nodo 1: Canal 4 Nodo 2: Canal 9
Seguridad	WPA2
Protocolo de interfaz Southbound	Openflow v1.3
Protocolo de interfaz Northbound	API RESTful
<b>Componentes de software</b>	
Controlador SDN	Ryu
Firmware de APs inalámbricos	OpenWrt con módulo OVS
<b>Componentes de Hardware</b>	
APs inalámbricos	Raspberry Pi 3B

### 3.2.1. Topología lógica de red SDWN

La **Figura 17** muestra la distribución del direccionamiento IP en la red. Esta distribución toma en cuenta tanto las capas de infraestructura y control, como los dispositivos que se conectarán a la red. El segmento de red empleado para toda la topología es 192.168.15.0/24. El direccionamiento entre el controlador y los puntos de acceso (APs) se destaca en color azul y es en donde se gestiona el protocolo OpenFlow. Por otro lado, el direccionamiento asignado a las estaciones de la red inalámbrica se indica en color marrón. Cabe destacar que los servidores operarán con direcciones IP estáticas, mientras que los demás dispositivos serán clientes DHCP.

**Figura 17**  
Topología lógica de red SDWN.



### 3.2.2. Instalación de controlador Ryu

Aunque el trabajo presentado por Moncayo (2023) ejecuta el controlador Ryu en una máquina virtual con sistema operativo Debian 11, en este trabajo se prefiere emplear Ubuntu Desktop 22.04 LTS. Esta elección se basa en que, al intentar emplear la versión más reciente, Debian 12, se han encontrado dificultades para instalar los paquetes de sistema necesarios para la funcionalidad de Ryu. Los requerimientos óptimos para el funcionamiento de la máquina virtual se detallan en la **Tabla 9**.

**Tabla 9**

*Requerimientos de máquina virtual para controlador.*

Requerimiento	Especificación
Memoria RAM	4 GB (mínimo)
Almacenamiento	50 GB (recomendado)
Arquitectura	64 bits
Procesador	2 núcleos

La instalación del controlador puede realizarse de dos maneras: a través del repositorio de GitHub para obtener el código fuente, o utilizando pip, el gestor de paquetes de Python. Sin embargo, independientemente del método, es fundamental realizar una instalación previa de los complementos del sistema. Esta medida resulta crucial para prevenir errores, tal como se muestra en la **Figura 18**, y garantizar un funcionamiento óptimo del entorno de Ryu. Estos complementos se describen en la **Tabla 10**.

**Figura 18**

*Fallo en la inicialización del controlador Ryu.*

```

karla@ryu:~$ ryu-manager
Traceback (most recent call last):
  File "/usr/local/bin/ryu-manager", line 5, in <module>
    from ryu.cmd.manager import main
  File "/usr/local/lib/python3.10/dist-packages/ryu/cmd/manager.py", line 33, in
<module>
    from ryu.app import wsgi
  File "/usr/local/lib/python3.10/dist-packages/ryu/app/wsgi.py", line 109, in <
module>
    class _AlreadyHandledResponse(Response):
  File "/usr/local/lib/python3.10/dist-packages/ryu/app/wsgi.py", line 111, in _
AlreadyHandledResponse
    from eventlet.wsgi import ALREADY_HANDLED
ImportError: cannot import name 'ALREADY_HANDLED' from 'eventlet.wsgi' (/usr/loc
al/lib/python3.10/dist-packages/eventlet/wsgi.py)
karla@ryu:~$

```

**Tabla 10**  
Prerrequisitos para la instalación de Ryu.

Paquete	Funcionalidad con controlador Ryu
<b>Complementos de sistema</b>	
gcc	Permite compilar extensiones de Python escritas en el lenguaje C.
libffi-dev	Esencial para la interoperabilidad entre lenguajes de programación.
libssl-dev	Facilita la comunicación segura a través de SSL <sup>18</sup> . Es utilizado por OpenFlow.
libxml2-dev	Es la biblioteca para procesamiento de XML <sup>19</sup> , utilizado por Ryu para manipulación de mensajes OpenFlow o para procesar datos en formato XML.
libxslt1-dev	Es la biblioteca para transformaciones XSLT <sup>20</sup> , puede ser utilizado por Ryu para transformaciones de datos.
zlib1g-dev	Proporciona funciones para comprimir y descomprimir datos, útil para algunas bibliotecas de Python.
<b>Complementos de Python</b>	
python3	Interprete de Python utilizado por el controlador Ryu.
python3-pip	Gestor de paquetes de Python utilizado para instalar y administrar bibliotecas de Python.
python3-eventlet	Framework de red utilizado en el controlador Ryu para implementar la comunicación asíncrona.
python-dev-is-python3	Requerido para compilar e instalar extensiones de Python.

*Nota.* Adoptado de Ryu Documentation (2014) y Python Documentation (2024).

Los paquetes mencionados se instalan mediante el comando proporcionado a continuación, y la **Figura 19** muestra la ejecución del mismo.

```
$ sudo apt install -y gcc libffi-dev libssl-dev libxml2-dev libxslt1-dev
zlib1g-dev python3 python3-pip python3-eventlet python-dev-is-python3
```

<sup>18</sup> SSL (Secure Sockets Layer), es un protocolo de seguridad en las comunicaciones de Internet (Kaspersky, 2024).

<sup>19</sup> XML (Extensible Markup Language), es un lenguaje de marcado flexible diseñado para almacenar y transmitir datos de manera legible tanto para humanos como para sistemas informáticos (Loshin et al., 2024).

<sup>20</sup> XSLT (Extensible Stylesheet Language Transformations), es un lenguaje de transformación utilizado para cambiar la estructura y presentación de documentos XML a otros formatos, como HTML (Sheldon, 2024).

### Figura 19

*Dependencias de Python3 necesarias para controlador Ryu.*

```

karla@ryu: ~
karla@ryu:~$ sudo apt install -y gcc libffi-dev libssl-dev libxml2-dev libxslt1-
dev zlib1g-dev python3 python3-pip python3-eventlet python-dev-is-python3
[sudo] password for karla:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho

```

Una vez finalizado el proceso de instalación, se verifica el mismo mediante el comando `--version`, precedido por el nombre del paquete que se desee comprobar. Por ejemplo, para verificar la instalación de Python3 y pip3, se puede utilizar los comandos `python3 --version` y `pip3 --version`, respectivamente. La **Figura 20** muestra la salida esperada de estos comandos, confirmando que la instalación fue exitosa.

### Figura 20

*Verificación de paquetes instalados.*

```

karla@ryu:~$ python3 --version
Python 3.10.12
karla@ryu:~$ pip3 --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
karla@ryu:~$

```

Posteriormente, la instalación del controlador Ryu se realiza utilizando el segundo método, el cual implica el uso del comando `pip3`, el cual es parte de Python3<sup>21</sup>. De este modo el comando de instalación es `sudo pip3 install ryu`. La **Figura 21** evidencia la instalación de Ryu.

### Figura 21

*Instalación de Ryu con pip3.*

```

karla@ryu: ~
karla@ryu:~$ sudo pip3 install ryu
Collecting ryu
  Downloading ryu-4.34.tar.gz (1.1 MB)
----- 1.1/1.1 MB 1.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: eventlet!=0.18.3,!=0.20.1,!=0.21.0,!=0.23.0,>=0.18.2 in /usr/lib/python3/dist-packages (from ryu) (0.30.2)
Collecting msgpack>=0.3.0
  Downloading msgpack-1.0.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (385 kB)
----- 385.1/385.1 KB 1.4 MB/s eta 0:00:00

```

<sup>21</sup> Python3 representa la versión más actualizada del lenguaje Python. Al utilizar bibliotecas y dependencias de Python, es esencial tener en cuenta la versión específica, ya que Python3 no es compatible con versiones anteriores, como Python2.

Adicionalmente, Ryu ofrece una serie de programas de ejemplo proporcionados por su comunidad de desarrolladores. Para acceder a ellos, es necesario clonar el repositorio de GitHub, tal como se muestra en la **Figura 22**.

### Figura 22

*Clonación de ejemplos de Python para controlador Ryu.*

```

karla@ryu: ~
karla@ryu:~$ git clone https://github.com/faucetsdn/ryu.git
Clonando en 'ryu'...
remote: Enumerating objects: 26506, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 26506 (delta 1), reused 4 (delta 1), pack-reused 26499
Recibiendo objetos: 100% (26506/26506), 13.95 MiB | 1.13 MiB/s, listo.
Resolviendo deltas: 100% (19160/19160), listo.
karla@ryu:~$

```

Finalmente, para iniciar el controlador Ryu, se utiliza el programa de ejemplo "simple\_switch\_15", obtenido del repositorio clonado previamente. El comando para llevar a cabo esta acción es `ryu-manager --verbose ryu.app.simple_switch_15`. En la **Figura 23** se muestra esta inicialización, donde se observa que después de iniciar el controlador, se activa el proceso "ofp\_handler", que es el encargado de manejar los eventos de OpenFlow.

### Figura 23

*Inicialización de controlador Ryu.*

```

karla@ryu:~/ryu$ ryu-manager --verbose ryu.app.simple_switch_15
loading app ryu.app.simple_switch_15
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_15 of SimpleSwitch15
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK SimpleSwitch15
  CONSUMES EventOFPPacketIn
  CONSUMES EventOFPSwitchFeatures
BRICK ofp_event
  PROVIDES EventOFPPacketIn TO {'SimpleSwitch15': {'main'}}

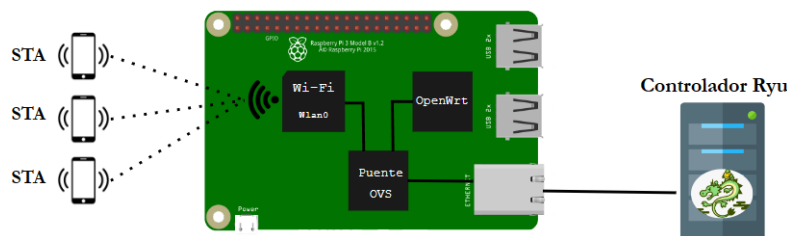
```

### 3.2.3. Configuración de APs programables

Para equipar el hardware Raspberry Pi 3B con la capacidad de funcionar como puntos de acceso programables, el primer paso consiste en instalar el firmware OpenWrt. Posteriormente, sobre este sistema, se instala el módulo Open Virtual Switch (OVS). Este módulo permite la creación de un puente virtual entre las interfaces cableadas e inalámbricas, y su administración a través de la programación establecida en el controlador utilizando el protocolo OpenFlow, tal como se ilustra en la **Figura 24**.

**Figura 24**

*Diagrama de conexión de OpenWrt y OVS en RPi 3B.*



#### 3.2.3.1. Instalación de firmware OpenWrt en RPi 3B

La escritura del firmware OpenWrt se realiza mediante el software Raspberry Pi Imager<sup>22</sup>. En este sentido, OpenWrt ofrece archivos de firmware para una amplia gama de hardware, disponible en su sitio oficial<sup>23</sup>. En este caso particular se descarga el instalador para Raspberry Pi 3B, como se muestra en la **Figura 25**.

**Figura 25**

*Descarga de firmware OpenWrt para Raspberry Pi3B.*

Model	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi	Raspberry Pi 2	Raspberry Pi 2	Raspberry Pi 3	Raspberry Pi 3	Raspberry Pi 3
Version	A	B	B+	Zero	Zero W	B 1.0/1.1	B 1.2	B	B+	Compute Module 3
Supported Current Rel	23.05.0	23.05.0	23.05.0	23.05.0	23.05.0	23.05.0	23.05.0	23.05.0	23.05.0	23.05.0
Firmware OpenWrt Install URL	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>	<a href="#">Factory image</a>

<sup>22</sup> Raspberry Pi Imager es una herramienta de software proporcionada por la Fundación Raspberry Pi para facilitar la instalación de sistemas operativos en tarjetas de memoria para dispositivos Raspberry Pi.

<sup>23</sup> La URL del sitio oficial es la siguiente: [https://openwrt.org/es/toh/views/toh\\_fwdownload](https://openwrt.org/es/toh/views/toh_fwdownload)

El almacenamiento utilizado en RPi 3B es una tarjeta de memoria microSD de 16 GB. Esta tarjeta se inserta en un adaptador de memoria microSD y se conecta a una computadora para prepararla adecuadamente antes de proceder con la instalación del firmware (ver **Figura 26**). El proceso de preparación consta de dos pasos esenciales: (i) verificar que la computadora reconozca sin inconvenientes la tarjeta microSD y (ii) formatear la tarjeta para asignar todo el espacio disponible a la instalación del firmware.

**Figura 26**

*Uso de adaptador para lectura de tarjeta microSD en PC.*



Dentro del entorno de Raspberry Pi Imager, es esencial realizar tres selecciones; (i) el modelo de hardware al que se destinará el firmware, (ii) el sistema operativo correspondiente al archivo descargado y descomprimido previamente, (iii) la tarjeta microSD en la que se efectuará la escritura y posteriormente se pulsa el botón “siguiente” (ver **Figura 27**).

Para la escritura del firmware, no se requiere ninguna configuración adicional; se utilizan los ajustes predeterminados. Y tras un breve periodo, este proceso se completará y aparecerá un mensaje indicando que ya es seguro retirar la tarjeta microSD. El paso siguiente consiste en insertar la tarjeta en la Raspberry Pi 3B y encenderla.



**Figura 27**  
*Escritura de Firmware con Raspberry Pi Imager.*



A continuación, para administrar el sistema OpenWrt instalado en el dispositivo RPi 3B se emplea el acceso remoto. Este proceso requiere la conexión de la Raspberry a una computadora utilizando un cable de red directo (Ver **Figura 28**). Durante este proceso, es necesario configurar una dirección IPv4 estática en la interfaz Ethernet de la PC, la cual debe estar en la misma subred que la Raspberry, específicamente en la 192.168.1.0/24, con esto se establece una comunicación punto a punto entre los dispositivos y es posible acceder a la Raspberry de manera remota. Cabe destacar que la dirección IP predeterminada de OpenWrt es 192.168.1.1/24.

**Figura 28**  
*Configuración de sistema OpenWrt mediante acceso remoto.*



Para el acceso remoto desde la computadora, se emplea la línea de comandos de Windows (CMD) junto con el servicio SSH. Las credenciales de acceso predeterminadas son; usuario “root” y contraseña en blanco. Así, para iniciar sesión, se emplea el comando: `ssh root@192.168.1.1`. Y tras acceder al sistema del dispositivo, es necesario establecer una

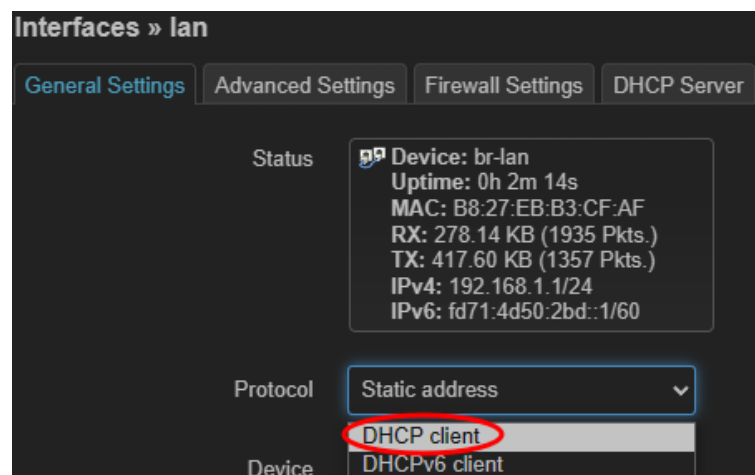


### 3.2.3.2. Instalación de módulo Open Virtual Switch en el sistema OpenWrt

Para proceder con la instalación del módulo OVS, es esencial que la Raspberry tenga acceso a Internet. Para lograr esto, se configura la interfaz eth0 en modo cliente DHCP. Este ajuste se puede realizar a través de la interfaz web, para ello hay que desplegar el menú 'Network' y seleccionar la opción 'Interfaces', a continuación, seleccionar y editar la interfaz LAN como se ilustra en la **Figura 31**. Posteriormente, se conecta la Raspberry al punto de acceso proporcionado por el proveedor de la red local, lo que garantiza la conexión a Internet.

**Figura 31**

*Configuración de interfaz LAN en modo cliente DHCP.*



Con la Raspberry ya conectada a Internet, se puede comenzar con la actualización de todos los paquetes del sistema, este es un proceso necesario para evitar problemas y asegurar que el sistema funcione de manera eficiente y segura. Para hacerlo, se utiliza el comando `opkg update`. La **Figura 32** evidencia este proceso.

**Figura 32**

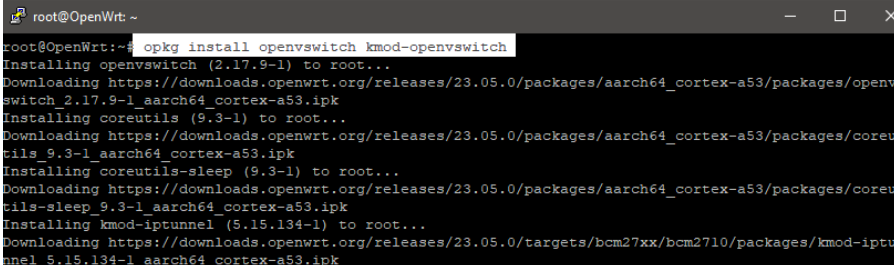
*Actualización del sistema OpenWrt.*

```
root@OpenWrt:~# opkg update
Downloading https://downloads.openwrt.org/releases/23.05.0/targets/bcm27xx/bcm27
10/packages/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_core
Downloading https://downloads.openwrt.org/releases/23.05.0/targets/bcm27xx/bcm27
10/packages/Packages.sig
Signature check passed.
Downloading https://downloads.openwrt.org/releases/23.05.0/packages/aarch64_cort
ex-a53/base/Packages.gz
Updated list of available packages in /var/opkg-lists/openwrt_base
Downloading https://downloads.openwrt.org/releases/23.05.0/packages/aarch64_cort
ex-a53/base/Packages.sig
Signature check passed.
```

Para la instalación de OVS se requieren dos paquetes; `openvswitch` y `kmod-openvswitch`. El primero contiene el software principal de Open vSwitch, mismo que proporciona las funcionalidades básicas de conmutación y enrutamiento. Mientras que el segundo incluye los módulos del kernel<sup>24</sup> necesarios para el funcionamiento adecuado de OVS con el sistema operativo subyacente. La **Figura 33** muestra la instalación de los paquetes mencionados.

Posteriormente, se verifica que OVS se haya instalado correctamente. Para ello se comprueba el servicio `ovs-vswitchd`, como se evidencia en la **Figura 34**, donde se visualiza la instalación de OVS ha sido exitosa y que la versión instalada es 2.17.9. Además, se analiza el servicio `ovs-ofctl`, el cual utiliza el protocolo OpenFlow en conjunto con OVS. En esta instancia, se observa que OVS es compatible con versiones de OpenFlow que van desde la 1.0 hasta la 1.5.

**Figura 33**  
*Instalación de Open Virtual Switch.*

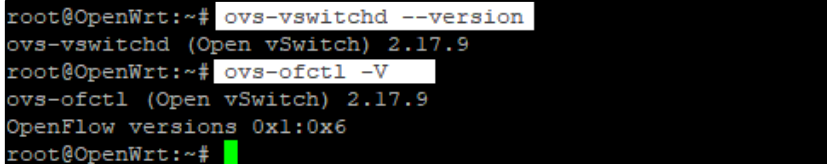


```

root@OpenWrt:~# opkg install openvswitch kmod-openvswitch
Installing openvswitch (2.17.9-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/packages/aarch64_cortex-a53/packages/openvswitch_2.17.9-1_aarch64_cortex-a53.ipk
Installing coreutils (9.3-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/packages/aarch64_cortex-a53/packages/coreutils_9.3-1_aarch64_cortex-a53.ipk
Installing coreutils-sleep (9.3-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/packages/aarch64_cortex-a53/packages/coreutils-sleep_9.3-1_aarch64_cortex-a53.ipk
Installing kmod-iptunnel (5.15.134-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/targets/bcm27xx/bcm2710/packages/kmod-iptunnel_5.15.134-1_aarch64_cortex-a53.ipk

```

**Figura 34**  
*Versión de OVS y versiones de OpenFlow soportadas por el mismo.*



```

root@OpenWrt:~# ovs-vswitchd --version
ovs-vswitchd (Open vSwitch) 2.17.9
root@OpenWrt:~# ovs-ofctl -V
ovs-ofctl (Open vSwitch) 2.17.9
OpenFlow versions 0x1:0x6
root@OpenWrt:~#

```

<sup>24</sup> Kernel o núcleo es el componente central de un sistema operativo que actúa como intermediario entre el software de la computadora y el hardware (Soto, 2020).

### 3.2.3.3. Creación de puente con Open Virtual Switch

Como se detalla en la **Sección 3.2.3**, el módulo OVS se utiliza para crear un puente virtual que involucre todas las interfaces de red, tanto cableadas como inalámbricas, facilitando así su administración centralizada a través del controlador mediante el protocolo OpenFlow. Sin embargo, dado que el hardware utilizado en este trabajo solo cuenta con una interfaz Ethernet, se procede a instalar los paquetes necesarios para reconocer los adaptadores de USB 2.0 a Ethernet, que permitirán la conexión con los nodos de red propuestos en la topología (ver **Figura 17**). Este proceso debe llevarse a cabo en los puntos de acceso 1 y 3. Tras esta implementación, se procederá a la creación del puente virtual entre las interfaces requeridas.

La **Figura 35** muestra la instalación de los drivers que soporten adaptadores de red USB, en donde los paquetes `kmod-usb2` y `kmod-usb-net-rtl8152` son necesarios. El primer paquete proporciona soporte para dispositivos USB2.0, mientras que el segundo está diseñado específicamente para soportar adaptadores de red USB que utilizan el chip RTL8152, así, en el caso de contar con un adaptador que emplee un chip diferente, será necesario buscar el paquete que brinde soporte a dicho chip. Además, cabe destacar que durante la instalación es esencial que el adaptador esté conectado para que el sistema pueda reconocerlo.

**Figura 35**

*Instalación de paquete para reconocer adaptador USB2.0 a Ethernet.*

```

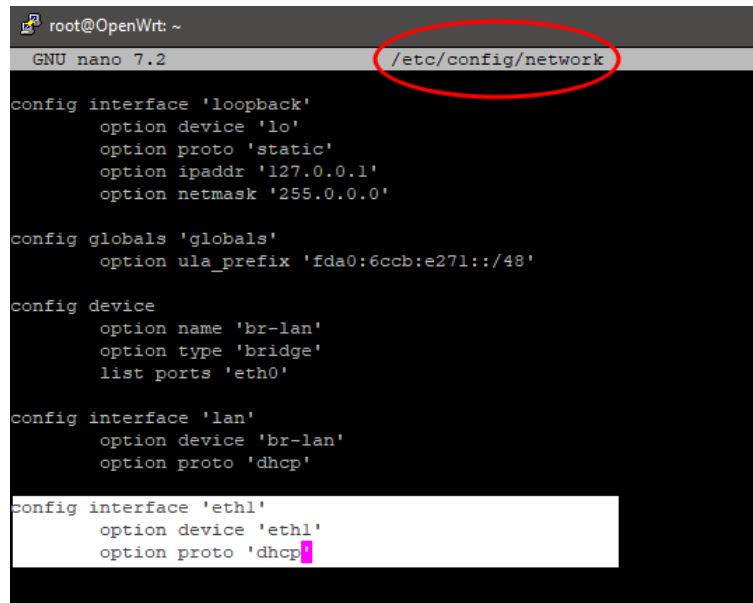
root@OpenWrt:~# opkg install kmod-usb2
Installing kmod-usb2 (5.15.134-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/targets/bcm27xx/bcm2710/packages/kmod-usb2_5.15.134-1_aarch64_cortex-a53.ipk
Installing kmod-usb-ehci (5.15.134-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/targets/bcm27xx/bcm2710/packages/kmod-usb-ehci_5.15.134-1_aarch64_cortex-a53.ipk
Configuring kmod-usb-ehci.
Configuring kmod-usb2.
root@OpenWrt:~# opkg install kmod-usb-net-rtl8152
Installing kmod-usb-net-rtl8152 (5.15.134-1) to root...
Downloading https://downloads.openwrt.org/releases/23.05.0/targets/bcm27xx/bcm2710/packages/kmod-usb-net-rtl8152_5.15.134-1_aarch64_cortex-a53.ipk
Installing r8152-firmware (20230804-1) to root...

```

Después, se procede a editar el archivo de configuración network que se ubica en la ruta `/etc/config`. En este archivo se crea las interfaces que funcionarán mediante los adaptadores de USB a Ethernet configurados previamente. La **Figura 36** muestra la creación de la interfaz

eth1, una de las interfaces configuradas para este propósito. Para finalizar el proceso de creación de interfaces de red, es necesario reiniciar el sistema, esto se logra con el comando `reboot`.

**Figura 36**  
*Edición de archivo de configuración de red.*



```

root@OpenWrt: ~
GNU nano 7.2 /etc/config/network

config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fda0:6ccb:e271::/48'

config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth0'

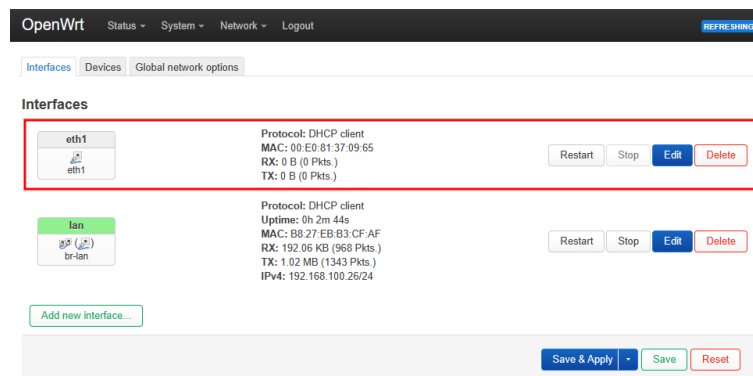
config interface 'lan'
    option device 'br-lan'
    option proto 'dhcp'

config interface 'eth1'
    option device 'eth1'
    option proto 'dhcp'
  
```

**Nota.** La figura muestra la configuración de `eth1` en el AP3, no obstante, para AP1 es necesario configurar las interfaces `eth1` y `eth2` para las cuales se emplea las mismas líneas de configuración empleadas con `eth1`.

Seguidamente, se verifica que la creación de interfaces se haya completado, para ello desde la interfaz web se ingresa en la sección de interfaces y si el proceso se ha llevado a cabo correctamente, la interfaz creada debería aparecer tal como se muestra **Figura 37**.

**Figura 37**  
*Interfaz eth1 habilitada.*



Después, se procede con la creación del puente con OVS. Para ello se comienza habilitando el servicio openvswitch como se muestra en la **Figura 38**. Posteriormente, la **Figura 39** muestra la creación del puente virtual con el comando `ovs-vsctl add-br sdwlan0`, en este comando el parámetro `'sdwlan0'` es el nombre o identificador que se asigna al puente. Además, se agrega el ID al puente con el comando `ovs-vsctl set bridge br-sdwlan0 other-config:datapath-id=0000000000000002`, donde la variable `'datapath-id=0000000000000002'` representa el ID específico del puente.

### Figura 38

*Inicialización de OVS.*

```
root@OpenWrt:~# /etc/init.d/openvswitch start
root@OpenWrt:~# /etc/init.d/openvswitch enable
root@OpenWrt:~# ovs-ctl status
ovsdb-server is running with pid 2104
ovs-vswitchd is running with pid 2117
```

### Figura 39

*Creación de puente virtual con OVS.*

```
~# ## Creacion de bridge OVS ##
~# ovs-vsctl add-br br-sdwlan0
~# ovs-vsctl set bridge br-sdwlan0 other-config:datapath-id=0000000000000002
```

Más adelante, se agregan las interfaces de red al puente creado. En este caso se agregan la interfaz inalámbrica `'phy0-ap0'` y la interfaz Ethernet `'eth1'`. Se verifica esta configuración con el comando `ovs-vsctl show`, este comando proporciona una visión detallada de la configuración del puente virtual, permitiendo verificar que las interfaces se han añadido de manera exitosa, como se visualiza en la **Figura 40**.

**Figura 40***Interfaces agregadas a puente OVS.*

```

root@OpenWrt:~# ovs-vsctl add-port swlan0 phy0-ap0
ovs-vsctl: cannot create a port named phy0-ap0 because a port named phy0-ap0 already exists on bridge swlan0
root@OpenWrt:~# ovs-vsctl add-port swlan0 eth1
root@OpenWrt:~# ovs-vsctl show
6f7d68f2-ebf1-4aec-942f-c5f976e160b7
    Bridge swlan0
        Port swlan0
            Interface swlan0
                type: internal
        Port phy0-ap0
            Interface phy0-ap0
        Port eth1
            Interface eth1
    ovs version: "2.17.9"

```

### 3.2.3.4. Establecimiento de comunicación entre el controlador y los APs

Posterior a la creación del puente con OVS, se procede a realizar pruebas de comunicación entre el controlador y los Puntos de Acceso (APs). Este proceso requiere la activación del puente virtual, que a partir de este punto será configurado y administrado como una interfaz de red. Para llevar a cabo esta tarea, se utiliza el comando `ip link set swlan0 up` como se muestra en la **Figura 41**.

También, se asigna una dirección a la interfaz mencionada. En este caso, teniendo en cuenta la topología lógica de la red SDWN (ver **Figura 17**), se aplica la dirección estática 192.168.15.11/24. Este proceso se lleva a cabo desde la interfaz web. Para ello, se despliega el menú Network y en la sección de interfaces, se edita la interfaz 'swlan0', asignándole la dirección previamente mencionada. La **Figura 41** muestra el direccionamiento asignado a la interfaz.

**Figura 41***Habilitación y direccionamiento de interfaz bridge creada.*

```

root@OpenWrt:~# ip link set swlan0 up
root@OpenWrt:~# ifconfig
swlan0  Link encap:Ethernet  HWaddr 00:E0:81:37:09:65
        inet addr:192.168.15.11  Bcast:192.168.15.255  Mask:255.255.255.0
        inet6 addr: fe80::2e0:81ff:fe37:965/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:14 errors:0 dropped:0 overruns:0 carrier:0

```

A continuación, se establece la conexión con el controlador, tal como se muestra en la **Figura 42**. Para ello se emplea el comando `ovs-vsctl set-controller swlan0`



tcp:192.168.15.10:6653. Este comando activa la comunicación entre el controlador y el puente, dicha comunicación se establece a la dirección IP del controlador y en el puerto 6653/TCP, que corresponde al puerto de OpenFlow. Asimismo, mediante el comando `ovs-vsctl set bridge sdwlan0 protocolos=OpenFlow13`, se establece el protocolo que el puente `sdwlan0` debe usar para comunicarse con el controlador. En este caso, se especifica el protocolo OpenFlow v1.3.

#### Figura 42

*Habilitar comunicación de puente OVS con controlador mediante OpenFlow.*

```
root@OpenWrt:~# ovs-vsctl set-controller br-sdwlan0 tcp:192.168.15.10:6653
```

Para verificar la conexión entre el controlador y los APs, desde el controlador se emplea el comando `ryu-manager simple_switch_13.py` para ejecutar el ejemplo `simple_switch_13`. Este ejemplo proviene del repositorio “ryu”, clonado en la **Sección 3.2.2**. Así, la **Figura 43** muestra la salida generada por la ejecución de dicho ejemplo, donde el mensaje “connected socket” indica que se ha establecido una conexión exitosa con los cuatro APs de la red, en este punto es importante destacar que el puerto empleado por los APs es asignado de forma aleatoria.

#### Figura 43

*Ejecución de ejemplo para probar comunicación entre APs y controlador.*

```
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK SimpleSwitch15
CONSUMES EventOFPPacketIn
CONSUMES EventOFPSwitchFeatures
connected socket:<eventlet.greenio.base.GreenSocket object at 0x7fd8ec909ea0> address:('192.168.15.13', 49248)
connected socket:<eventlet.greenio.base.GreenSocket object at 0x7fd8ec909a20> address:('192.168.15.14', 47264)
hello ev <ryu.controller.ofp_event.EventOFPHello object at 0x7fd8ec90ba90>
move onto config mode
EVENT ofp_event->SimpleSwitch15 EventOFPSwitchFeatures
switch features ev version=0x6,msg_type=0x6,msg_len=0x20,xid=0x40ecbe7e,OFPSwitchFeatures(auxiliary_id=0,capabilities=0,n_tables=254)
move onto main mode
hello ev <ryu.controller.ofp_event.EventOFPHello object at 0x7fd8ec950280>
move onto config mode
EVENT ofp_event->SimpleSwitch15 EventOFPSwitchFeatures
switch features ev version=0x6,msg_type=0x6,msg_len=0x20,xid=0x562d9a7e,OFPSwitchFeatures(auxiliary_id=0,capabilities=0,n_tables=254)
move onto main mode
connected socket:<eventlet.greenio.base.GreenSocket object at 0x7fd8ec90b550> address:('192.168.15.11', 54096)
connected socket:<eventlet.greenio.base.GreenSocket object at 0x7fd8ec9505e0> address:('192.168.15.12', 52870)
EVENT ofp_event->SimpleSwitch15 EventOFPPacketIn
packet in 4 00:e0:99:00:11:08 ff:ff:ff:ff:ff:ff 9
hello ev <ryu.controller.ofp_event.EventOFPHello object at 0x7fd8ec908b80>
move onto config mode
```

Adicionalmente, para verificar la conexión desde la Raspberry, se ejecuta el comando `ovs-vsctl show`, la salida de este comando se ilustra en la **Figura 44**, donde se visualiza que el parámetro “`is_connected`” está establecido en “`true`”, confirmando así la comunicación entre estos dispositivos.

**Figura 44**  
Verificación de comunicación entre los APs y el controlador.

```

root@OpenWrt:~# ##### AP1 #####
root@OpenWrt:~# ovs-vsctl show
e63a69c0-28a5-4f4a-8ee3-74706c00b7f1
Manager "ptcp:6632"
Bridge br-sdwan0
  Controller "tcp:192.168.15.10:6653"
  is_connected: true
  Port eth1
  Interface eth1
  Port phy0-ap0
  Interface phy0-ap0
  Port br-sdwan0
  Interface br-sdwan0
  type: internal
  Port eth2
  Interface eth2
  Port br-lan
  Interface br-lan
  ovs_version: "2.17.9"
root@OpenWrt:~#

root@OpenWrt:~# ##### AP2 #####
root@OpenWrt:~# ovs-vsctl show
3801c652-3a97-40f5-a195-f17cbe21412a
Manager "ptcp:6632"
Bridge br-sdwan0
  Controller "tcp:192.168.15.10:6653"
  is_connected: true
  Port br-sdwan0
  Interface br-sdwan0
  type: internal
  Port br-lan
  Interface br-lan
  Port phy0-ap0
  Interface phy0-ap0
  ovs_version: "2.17.9"
root@OpenWrt:~#

root@OpenWrt:~# ##### AP3 #####
root@OpenWrt:~# ovs-vsctl show
3e603654-22a9-40fc-b431-d3f6ee75618e
Manager "ptcp:6632"
Bridge br-sdwan0
  Controller "tcp:192.168.15.10:6653"
  is_connected: true
  Port br-lan
  Interface br-lan
  Port br-sdwan0
  Interface br-sdwan0
  type: internal
  Port phy0-ap0
  Interface phy0-ap0
  Port eth1
  Interface eth1
  ovs_version: "2.17.9"
root@OpenWrt:~#

root@OpenWrt:~# ##### AP4 #####
root@OpenWrt:~# ovs-vsctl show
d45e8c75-9dbc-4ebl-abe8-c230b9cca746
Manager "ptcp:6632"
Bridge br-sdwan0
  Controller "tcp:192.168.15.10:6653"
  is_connected: true
  Port br-sdwan0
  Interface br-sdwan0
  type: internal
  Port phy0-ap0
  Interface phy0-ap0
  Port br-lan
  Interface br-lan
  ovs_version: "2.17.9"
root@OpenWrt:~#

```

Finalmente, en el controlador se ejecuta el software Wireshark para realizar una captura de tráfico de la comunicación establecida. El tráfico capturado se muestra en la **Figura 45**, donde es posible identificar tráfico que utiliza el protocolo OpenFlow, lo que confirma una comunicación efectiva entre el controlador y los AP.

**Figura 45**  
Protocolo OpenFlow en ejecución de ejemplo.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xc1e4cc33
2	2.068134	192.168.15.11	192.168.15.10	OpenFlow	74	Type: OFPT_ECHO_REQUEST
3	2.068751	192.168.15.10	192.168.15.11	OpenFlow	74	Type: OFPT_ECHO_REPLY
4	2.070492	192.168.15.11	192.168.15.10	TCP	66	40170 → 6633 [ACK] Seq=9 Ack=9 Win=502 Len=
5	3.090005	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xc1e4cc33
6	6.170101	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xc1e4cc33
7	7.071760	192.168.15.11	192.168.15.10	OpenFlow	74	Type: OFPT_ECHO_REQUEST
8	7.072433	192.168.15.10	192.168.15.11	OpenFlow	74	Type: OFPT_ECHO_REPLY

### 3.2.3.5. Configuración de red inalámbrica en APs

La comunicación entre las capas de control e infraestructura, presentada en la sección anterior, representa la mitad del camino en el establecimiento de la red SDWN. Es así que, para completar dicha configuración, en esta sección se realiza la comunicación entre la capa de infraestructura y las estaciones de trabajo. Esta comunicación es inalámbrica y emplea el protocolo IEEE 802.11n, específicamente en la banda de frecuencia 2.4 GHz. En este sentido, la distribución de canales inalámbricos para los APs se asigna de forma estratégica para evitar interferencias entre ellos, tal como se detalla en la **Tabla 11**.

**Tabla 11**

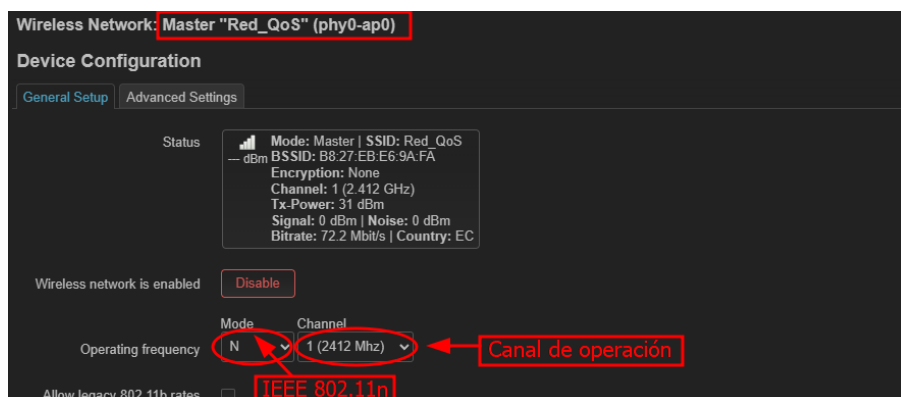
*Distribución de canales inalámbricos en APs.*

Access Point	Canal inalámbrico
AP1	CH1
AP2	CH4
AP3	CH7
AP4	CH10

La configuración inalámbrica se realiza a la interfaz phy0-ap0 del sistema OpenWrt. Para realizar esta tarea, se emplea la interfaz web de dicho sistema, en donde a través del menú Network se accede a la sección “Inalámbrico”, y principalmente se activa la conexión WLAN, posterior a ello se selecciona el protocolo inalámbrico y el canal de funcionamiento como lo ilustra la **Figura 46**.

**Figura 46**

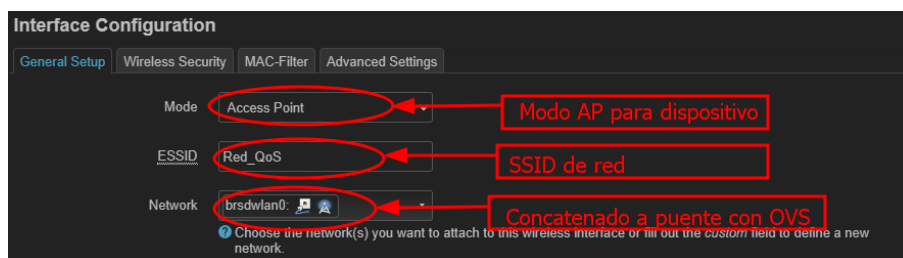
*Selección de protocolo inalámbrico y canal de funcionamiento.*



A continuación, se realizan las configuraciones generales (ver **Figura 47**). Inicialmente, se establece el modo de funcionamiento del sistema como punto de acceso. Luego, se define el nombre de la red en el campo ESSID<sup>25</sup>, en este punto es importante mencionar que se debe configurar el mismo identificador en todos los APs para garantizar la topología de red ESS. Finalmente se configura la red que se va a destinar el funcionamiento, en este caso es el puente OVS creado previamente.

**Figura 47**

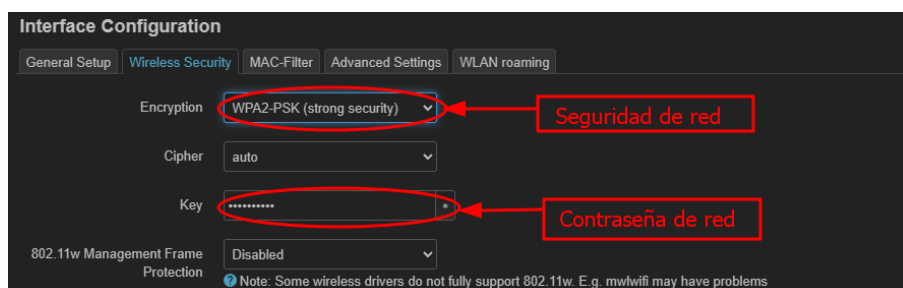
*Configuraciones de ESSID y conexión con bridge en la red.*



Más adelante, en la configuración de seguridad de la red se establece la encriptación WPA2-PSK con el cifrado por defecto, y se proporciona la clave de seguridad correspondiente, en este caso la clave configurada es 1234567890, como se señala en la **Figura 48**. Luego, para guardar las configuraciones descritas es necesario pulsar el botón ‘Save’.

**Figura 48**

*Configuración de seguridad WPA2 en la red.*

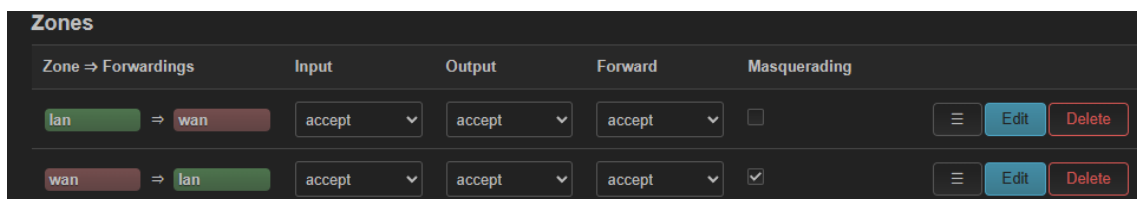


<sup>25</sup> ESSID hace referencia a Extended Service Set Identifier. Este término es utilizado en redes inalámbricas WLAN para referirse al nombre único asignado a un conjunto de dispositivos inalámbricos que se encuentran conectados a la misma red (Huawei Forums, 2021).

Para finalizar esta configuración es importante establecer las zonas de reenvío en el apartado de 'Firewall', ya que, sin esta configuración, es posible que se presenten dificultades en la asociación de las estaciones (STA) a los puntos de acceso. En este contexto, se configuran las políticas 'Aceptar' para regular tanto los paquetes que salen como los que ingresan a la red WLAN (ver **Figura 49**).

### Figura 49

*Configuración de firewall para admitir conexiones de estaciones a la red.*



A continuación, se verifica que los puntos de acceso se encuentren funcionando en el canal inalámbrico establecido. En primer lugar, se comprueba la comunicación entre todos los AP mediante la herramienta ping, como se muestra en la **Figura 50**. Posteriormente, desde el AP1, se realiza un escaneo de canales en la banda de 2.4 GHz. Para ello, se accede al apartado "Estado" y se selecciona la opción "Análisis de canales". Esto permite visualizar los SSID de las redes cercanas y el canal en el que están operando (ver **Figura 51**). En este caso, el nombre de la red inalámbrica de cada AP fue modificado a OpenWrt\_APx, para facilitar la identificación de cada AP y su canal de operación.

**Figura 50**

*Comunicación con todos de los APs de la red desde API.*

```

root@OpenWrt:~# ping 192.168.15.12 -c 4
PING 192.168.15.12 (192.168.15.12): 56 data bytes
64 bytes from 192.168.15.12: seq=0 ttl=64 time=1.112 ms
64 bytes from 192.168.15.12: seq=1 ttl=64 time=0.942 ms
64 bytes from 192.168.15.12: seq=2 ttl=64 time=0.811 ms
64 bytes from 192.168.15.12: seq=3 ttl=64 time=0.888 ms

--- 192.168.15.12 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.811/0.938/1.112 ms
root@OpenWrt:~# ping 192.168.15.13 -c 4
PING 192.168.15.13 (192.168.15.13): 56 data bytes
64 bytes from 192.168.15.13: seq=0 ttl=64 time=3.352 ms
64 bytes from 192.168.15.13: seq=1 ttl=64 time=0.806 ms
64 bytes from 192.168.15.13: seq=2 ttl=64 time=0.742 ms
64 bytes from 192.168.15.13: seq=3 ttl=64 time=0.802 ms

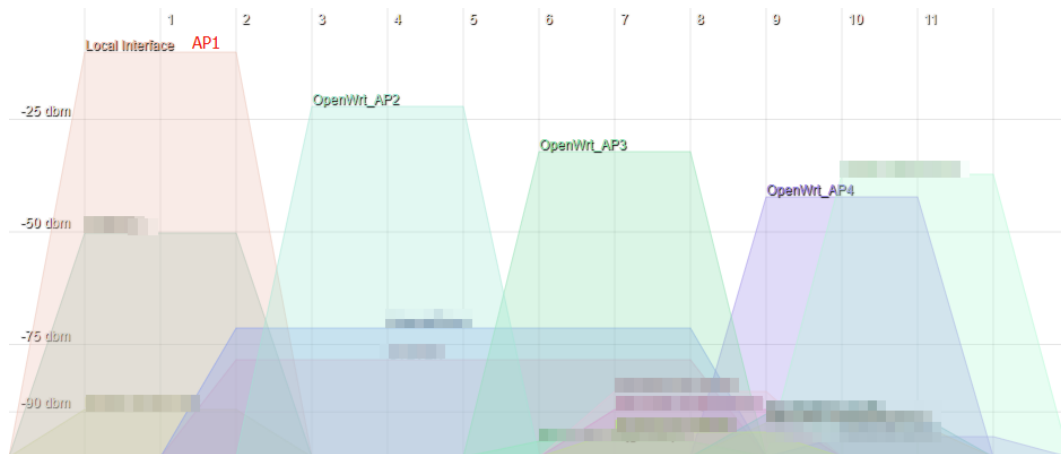
--- 192.168.15.13 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.742/1.425/3.352 ms
root@OpenWrt:~# ping 192.168.15.14 -c 4
PING 192.168.15.14 (192.168.15.14): 56 data bytes
64 bytes from 192.168.15.14: seq=0 ttl=64 time=1.434 ms
64 bytes from 192.168.15.14: seq=1 ttl=64 time=1.079 ms
64 bytes from 192.168.15.14: seq=2 ttl=64 time=1.086 ms
64 bytes from 192.168.15.14: seq=3 ttl=64 time=1.323 ms

--- 192.168.15.14 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.079/1.230/1.434 ms

```

**Figura 51**

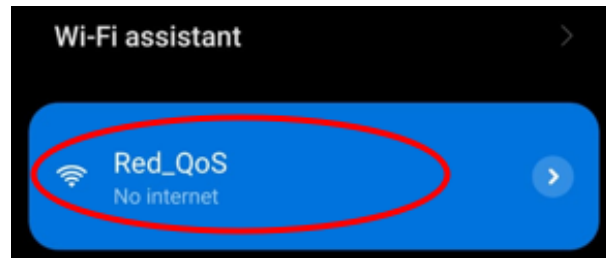
*Análisis de canales en la banda 2,4 GHz.*



Ahora, para comprobar que las configuraciones realizadas se hayan ejecutado correctamente, se procede a conectar una estación de trabajo a la red inalámbrica utilizando la clave de seguridad establecida; y luego de un breve periodo, se evidencia que el proceso de asociación entre la estación y el AP se ha concretado exitosamente. La **Figura 52** muestra el resultado de la conexión de la estación a la red.

**Figura 52**

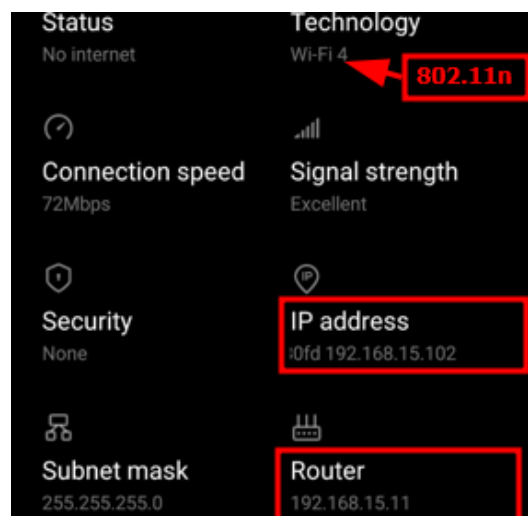
*Conexión a red SDWN desde smartphone Redmi Note 8.*



En este sentido, la **Figura 53** presenta los detalles de la red conectada, destacando que la estación (STA) ha accedido a la red a través del AP1, este hecho se evidencia debido a que la configuración de la puerta de enlace en la estación coincide con el direccionamiento (192.168.15.102) de dicho AP. Además, se observa que el protocolo inalámbrico 802.11n está operativo y en funcionamiento en esta conexión.

**Figura 53**

*Detalles de la conexión desde estación (STA).*

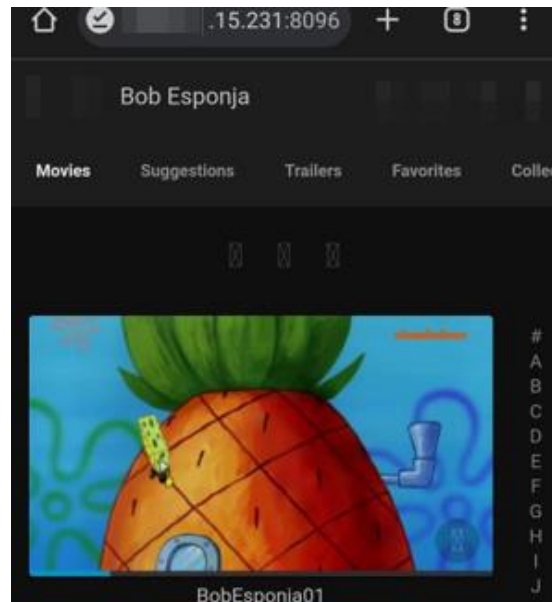


Para finalizar esta sección, desde la STA se accede al servicio de VOD que se encuentra en uno de los servidores de la red (para más detalles, consultar la configuración de servicios en el **Anexo A**). El acceso se efectúa utilizando la dirección IP del servidor y el puerto correspondiente, especificados como: `http://192.168.15.231:8096`. Así, la **Figura 54**

muestra el ingreso exitoso al servicio, demostrando la funcionalidad completa de la red, en concordancia con la topología de red establecida al inicio de este capítulo (ver **Figura 17**).

#### **Figura 54**

*Acceso a servidor VOD desde estación.*



### **3.3.Planteamiento y Aplicación de Políticas de QoS**

Este apartado se centra en el planteamiento y aplicación de QoS en la red SDWN. En la etapa de planteamiento se presenta la parametrización de los servicios presentes en la red y la adaptación de dichas características al entorno SDWN mediante una API RESTful. En este punto es importante destacar que la QoS se determina en función del modelo de Servicios Diferenciados (DiffServ) y la recomendación ITU-T Y.1541. Mientras que la etapa de aplicación implica la interacción directa con la API RESTful para la gestión de QoS desde el controlador Ryu hacia los puentes OVS establecidos en cada uno de los APs de la red.

#### **3.3.1. Planteamiento de QoS en la red SDWN**

En los entornos de red definidos por software (SDN), la gestión de la calidad de servicio (QoS) se lleva a cabo a través del controlador, el cual se encarga de distribuir las configuraciones de QoS a todos los dispositivos de la red. Esto permite una administración

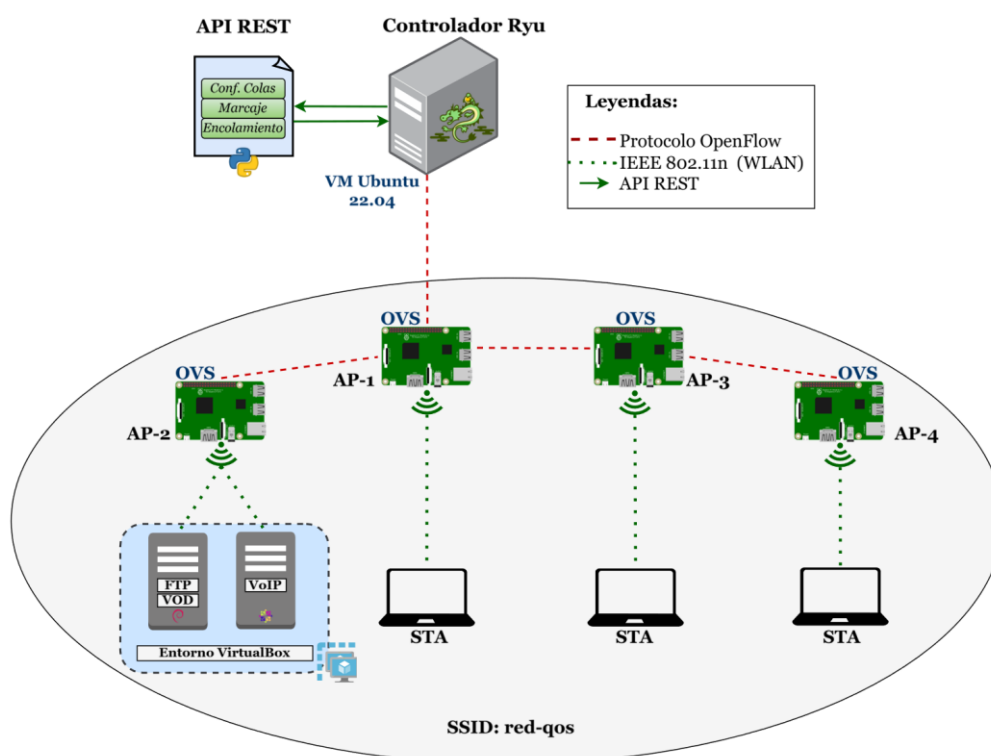


centralizada y mejora la eficiencia en los procesos de configuración de la red. Para este propósito, se utilizan APIs, generalmente basadas en la arquitectura REST, que facilitan la gestión y aplicación de políticas de QoS mediante solicitudes HTTP, empleando métodos como GET, POST, PUT y DELETE. Las respuestas a estas solicitudes se representan a través de códigos de estado que van del 100 al 599, los cuales indican si la operación se realizó con éxito, si ocurrió un error o si se requieren acciones adicionales.

Así, el despliegue de QoS para la red SDWN establecida en este trabajo se representa de color verde en la **Figura 55**. En esta figura se evidencia que la API RESTful administra los procesos de; configuración de colas, marcaje y clasificación de tráfico, los cuales constituyen las reglas de QoS que son gestionadas desde el controlador Ryu. Posteriormente, estas reglas son transmitidas a los APs a través del puente OVS, dado a que en la API se agregan librerías que permiten esta interacción, y es así como este proceso permite una distribución eficiente y uniforme de las políticas de QoS en toda la red.

**Figura 55**

*Funcionamiento de QoS sobre red SDWN.*



Por otra parte, la parametrización de los servicios; VoIP, VOD y FTP se lleva a cabo considerando la asociación entre las clases establecidas en la recomendación ITU-T Y.1541 y los PHBs definidos por DiffServ (ver **Tabla 6**). Es decir que, se asignan las características de las clases de servicio de ITU-T Y.1541 a los PHB de DiffServ para garantizar una implementación de calidad de servicio coherente, asegurando que cada clase reciba el tratamiento adecuado en términos de priorización y manejo del tráfico (ver **Sección 2.2.3**).

En este contexto, la **Tabla 12** se muestra la priorización de los servicios de red. Además, es importante resaltar que, de acuerdo con DiffServ, el ancho de banda total asignado no debe exceder el 70% del ancho de banda disponible, ya que el 30% restante se reserva para los procesos de red.

**Tabla 12**  
*Puertos de funcionamiento y prioridades en servicios de red SDWN.*

<b>Prioridad</b>	<b>Aplicación</b>	<b>Clase Y.1541 y PHB asociado</b>	<b>Porcentaje de AB asignado</b>	<b>Puerto/s</b>
<b>Alta</b>	VoIP	Clase 0 = EF	25%	UDP: [16384 - 32767] (RTP)
		Clase 1 = CS3	10%	UDP: 5060 (SIP)
<b>Media</b>	VOD	Clase 2 = AF32	15%	TCP: 8096
<b>Baja</b>	FTP	Clase 3 = AF23	10%	TCP: 21
<b>Por defecto</b>	Otras	Clase 5 = BE	S/N	Sin definir

Una vez establecida la priorización de servicios (**Tabla 12**), se procede a esquematizar la configuración de QoS sobre la red SDWN. Para ello, se analizan las características funcionales de la API RESTful, la cual es la encargada de recibir las configuraciones para la administración de las reglas de QoS, mismas abarcan los proceso de: (i) Configuración de colas, donde se establece el ancho de banda máximo y mínimo para cada cola; (ii) Marcaje, para establecer el valor del campo DSCP en función de la dirección IP destino y el puerto; (iii) Clasificación de tráfico, para definir la cola a la cual se le asignará al paquete, en función del valor en el campo DSCP.

Los procesos mencionados se configuran a través de solicitudes HTTP. Para ello se emplea la herramienta CURL (Cliente URL), la cual opera desde la línea de comandos y permite ejecutar peticiones a servidores y APIs. De acuerdo con Graham (2019), la sintaxis de estas solicitudes (request) se compone de cuatro partes:

- **Método HTTP:** Define la función de la solicitud.
- **Cuerpo:** Incluye los datos que se enviarán y es generalmente utilizado con los métodos POST y PUT.
- **Recurso o Endpoint:** Es la dirección URL que indica la ubicación y el identificador del recurso al cual se envía la solicitud.
- **Cabeceras:** Contienen metadatos sobre la solicitud.

Para aplicar esta sintaxis en la configuración de los tres procesos de QoS, se han definido dos recursos clave, uno de ellos se encarga de gestionar la configuración de colas, mientras que el otro administra las reglas de marcaje y clasificación de tráfico (ver **Tabla 13**). Además, se destaca que esta configuración se encuentra fundamentada en la documentación de Ryu (2014).

**Tabla 13**  
*Puntos finales para configuración de procesos de QoS.*

URL de Recurso	Descripción
<p>Controlador Ryu</p> <p>Identificador de recurso</p> <p>Recurso o Endpoint</p> <p>http://localhost:8080/qos/queue/switch-id</p>	<p>Especifica el punto final para la gestión de colas de QoS. Además, la variable {switch-id} es un parámetro que identifica el switch específico en el que se aplicará la configuración.</p>
<p>Controlador Ryu</p> <p>Identificador de recurso</p> <p>Recurso o Endpoint</p> <p>http://localhost:8080/qos/rules/switch-id</p>	<p>Indica el punto final para gestión de las reglas de marcaje y clasificación de tráfico. Y la variable {switch-id} especifica el switch al cual se dirige la configuración.</p>

**Nota.** Adoptado de Ryu (2014). En la columna izquierda, se destaca de color azul las partes que conforman la URL de cada recurso. Cabe destacar que la variable {switch-id} debe reemplazarse por el ID del dispositivo que se va a configurar.

De igual manera, la estructura para el cuerpo de las solicitudes POST y PUT varía en función del proceso de QoS que se esté implementando. Así, la **Tabla 14** detalla el formato JSON<sup>26</sup> necesario para cada proceso. Donde se visualiza que, en el caso de la configuración de colas, el cuerpo de la solicitud debe incluir el ancho de banda máximo y mínimo, así como el algoritmo que definirá la asignación de ancho de banda entre diferentes colas; en este caso, se emplea el algoritmo Hierarchical Token Bucket<sup>27</sup> (HTB). Para el marcaje de tráfico, se debe especificar la dirección IPv4 de destino, el protocolo, el puerto de operación y el valor del campo DSCP. Finalmente, para la clasificación de tráfico, se especifica el campo DSCP y el número de cola al que se asignará el tráfico correspondiente.

**Tabla 14**  
Formato JSON para las solicitudes de los procesos de QoS.

Proceso	Formato del cuerpo de la solicitud
<b>Configuración de colas</b>	<pre> {   "port_name": "&lt;phy0-ap0&gt;",   "type": "&lt;linux-htb&gt;",   "max-rate": "&lt;int&gt;",   "queues": [     {       "max_rate": "&lt;int&gt;",       "min_rate": "&lt;int&gt;"     }   ] } </pre> <p>     → Nombre de interfaz inalámbrica      → Algoritmo para la gestión de tráfico      → AB máximo de interfaz WLAN      → Definición de AB máximo y mínimo para cada cola   </p>
<b>Marcaje</b>	<pre> {   "match": {     "nw_dst": "&lt;Dir. IPv4 destino&gt;",     "nw_proto": "&lt;TCP, UDP, ICMP&gt;",     "tp_dst": "&lt;Puerto destino&gt;"   },   "actions": {     "mark": "&lt;Valor numérico de DSCP&gt;"   } } </pre> <p>     → Características del tráfico que se quiere identificar      → Dirección IP destino      → Protocolo de red      → Puerto destino      → Acciones para paquetes coincidentes      → Valor de campo DSCP   </p>

<sup>26</sup> JSON (JavaScript Object Notation), es un formato de texto utilizado para almacenar y compartir datos de manera legible tanto para personas como para máquinas (Erickson, 2024).

<sup>27</sup> HTB es un tipo de algoritmo de control de tráfico en Linux, permite especificar garantías de velocidad de bits por flujo y habilita el uso compartido de ancho de banda excedente entre flujos de la misma clase (Bosk et al., 2021).

<b>Clasificación de tráfico</b>	<pre> {   "match": {     "ip_dscp": "&lt;Valor DSCP&gt;",     "actions": {       "queue": "&lt;Número de cola&gt;"     }   } } </pre> <p>     Características del tráfico que se quiere identificar      Valor numérico de campo DSCP      Acciones para paquetes coincidentes      Cola asignada al tráfico coincidente   </p>
---------------------------------	---

*Nota.* Adoptado de Ryu (2014). Cabe señalar que los campos encerrados entre corchetes angulares deben sustituirse por los parámetros correspondientes de la configuración.

Adicionalmente, la **Tabla 15** ilustra ejemplos de la sintaxis que se debe emplear para realizar las diferentes solicitudes HTTP. Además, es importante resaltar que cada proceso requiere una solicitud independiente. Es decir, una solicitud no puede definir dos procesos simultáneamente, esto se realiza con el fin de manejar un entorno organizado y eficiente.

**Tabla 15**

*Ejemplos de solicitudes a los puntos finales definidos para QoS.*

**SOLICITUD GET:** Obtener las reglas de QoS del API.

```

curl -X GET http://localhost:8080/qos/rules/0000000000000001

```

Herramienta      Método HTTP      URL de Recurso o Endpoint

**SOLICITUD POST:** Crear una cola en el API.

```

curl -X POST -d '{"port_name": "br-sdwlan0", "type": "linux-htb", "max_rate": "70000000", "queues": [{"max_rate": "20000000", "min_rate": "5000000"}]}' http://localhost:8080/qos/queue/0000000000000001

```

Herramienta      Método HTTP      Cuerpo      URL de Recurso o Endpoint

**SOLICITUD PUT:** Actualizar la cola de tráfico marcado con DSCP=26, en el API.

```

curl -X PUT -d '{"match": {"ip_dscp": "26"}, "actions": {"queue": "2"}}' http://localhost:8080/qos/rules/0000000000000001

```

Herramienta      Método HTTP      Cuerpo      URL de Recurso o Endpoint

**SOLICITUD DELETE:** Eliminar todas las colas del API.

```

curl -X DELETE http://localhost:8080/qos/queue/0000000000000001

```

Herramienta      Método HTTP      URL de Recurso o Endpoint

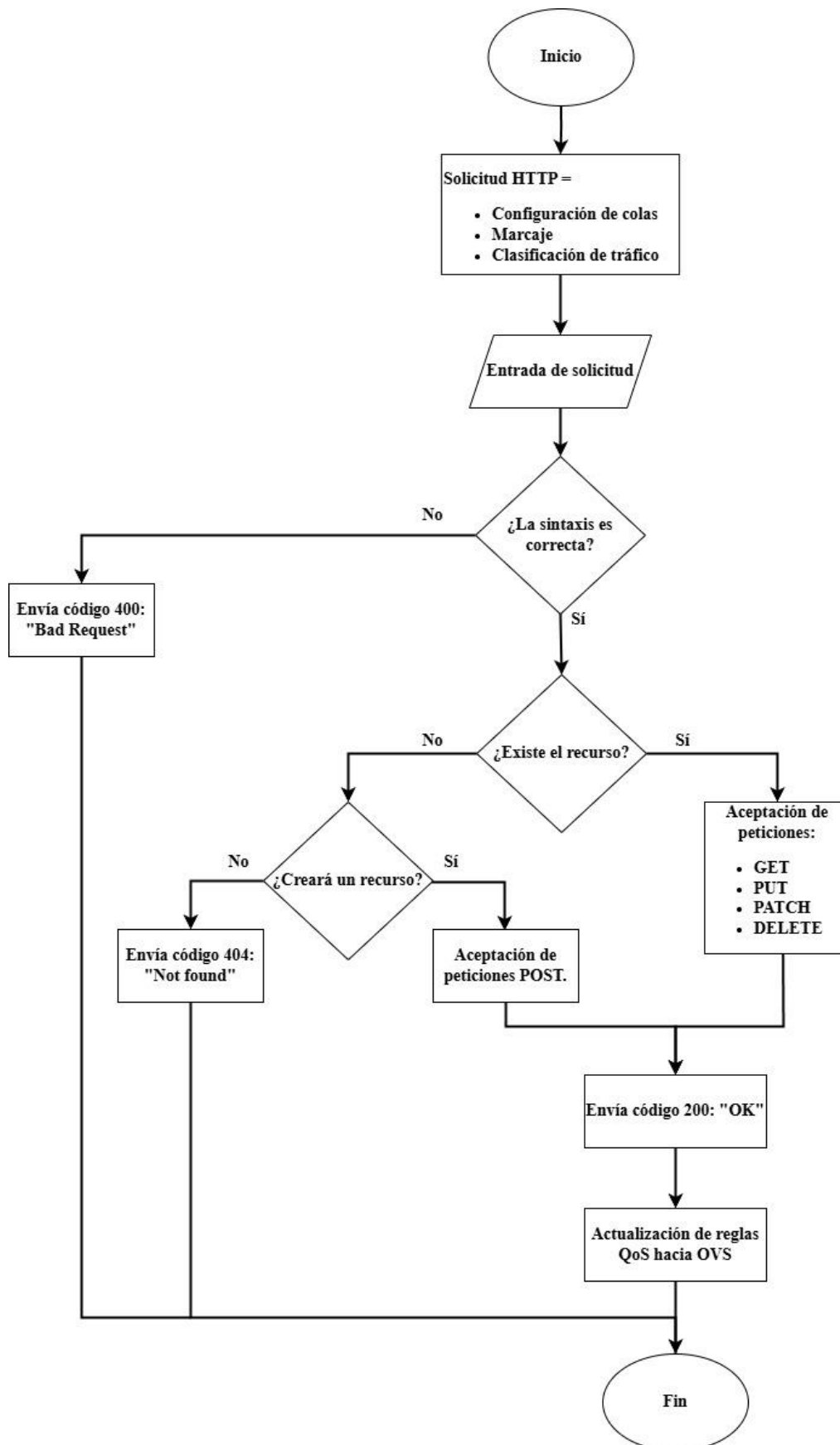
*Nota.* Tomado y adoptado de Ryu (2014). En la figura se observan ejemplos que no utilizan cabeceras, ya que no son necesarias para esta configuración.

En relación con la planificación de la configuración de QoS en la red, la **Figura 56** muestra el diagrama de flujo que ilustra el funcionamiento de la API RESTful para la gestión de QoS desde el controlador. En el diagrama se puede observar lo siguiente:

1. Al inicio, se define que los procesos de configuración de colas, marcaje y clasificación de tráfico serán administrados mediante solicitudes HTTP.
2. En la siguiente etapa, la API recibe una entrada que corresponde a una solicitud para gestionar las reglas de Calidad de servicio.
3. A continuación, la solicitud entra en una etapa de decisión donde se verifica que la sintaxis de la solicitud sea correcta, para continuar con el funcionamiento de la API. De lo contrario, se genera el código de estado 400, indicando un "Bad Request" y el proceso termina.
4. Si la sintaxis es correcta, la solicitud pasa a otra etapa de decisión para determinar si el recurso solicitado existe dentro de la API. Si el recurso existe, se acepta la petición y se envía el código 200, esto puede darse para las acciones de; GET, PUT y DELETE. Ya que la petición POST crea un recurso nuevo, proceso que se evalúa en la siguiente etapa.
5. La siguiente etapa de decisión permite validar si el usuario solicita crear un recurso nuevo mediante el método POST o, si el recurso llamado no existe. En el primer caso, la solicitud es aceptada, se envía el código 200 y la solicitud avanza a la siguiente etapa. En el segundo caso, se envía el código 404 indicando que el recurso no se encontró y el proceso termina.
6. Finalmente, las solicitudes que cumplan con las condiciones indicadas son configuradas dentro de la API y gestionadas hacia los puentes OVS de cada AP en la red.

**Figura 56**

Diagrama de flujo del funcionamiento de API RESTful para QoS.



### 3.3.2. Aplicación de QoS en la red

La configuración de QoS toma como punto de partida la API ‘rest\_qos.py’, descrita en la sección anterior. Esta interfaz permite la configuración de reglas y colas en cada uno de los AP de la red (ver **Figura 56**). Sin embargo, junto con esta, es necesario emplear los scripts ‘simple\_switch.py’ y ‘rest\_conf\_switch.py’, los cuales establecen la comunicación básica en la red. El primer script permite que los APs actúen como switches de capa dos, facilitando la conmutación de paquetes, mientras que el segundo permite la gestión centralizada de los APs desde el controlador (Ryu, 2014).

La API ‘rest\_qos.py’ establece la calidad de servicio en la red a través de la base de datos de OVS (OVSDB). De modo que, el controlador debe establecer la conexión hacia la OVSDB de cada AP. Una vez establecida esta conexión, será posible crear reglas y colas de QoS para gestionar la red inalámbrica de cada AP, siguiendo la sintaxis descrita en la **sección 3.3.1**.

#### 3.3.2.1. Establecimiento de conexión a OVSDB

La conexión a la base de datos de Open vSwitch (OVSDB) es un paso esencial para la gestión de QoS en la red. Para configurar adecuadamente esta conexión, es necesario establecer los parámetros ‘system-id’ y ‘datapath-id’ en cada uno de los puntos de acceso. Ambos parámetros son valores hexadecimales; el ‘system-id’ identifica de forma única al dispositivo en la red, mientras que el ‘datapath-id’ es el identificador utilizado en los procesos de OpenFlow. La **Tabla 16** muestra los valores establecidos en cada uno de los APs de la red. Asimismo, los comandos empleados para configurar estos parámetros se evidencian en la **Figura 57**.



**Tabla 16**  
Distribución de *system-id* y *datapath-id* en cada AP.

Puntos de Acceso	System-id	Datapath-id
AP1	000000000000000001	000000000000000001
AP2	000000000000000002	000000000000000002
AP3	000000000000000003	000000000000000003
AP4	000000000000000004	000000000000000004

**Figura 57**  
Configuración de *system-id* y *datapath-id* en AP1.

```
OpenWrt:~# ovs-vsctl set Open_vSwitch . external_ids:system-id=0000000000000001
OpenWrt:~# ovs-vsctl set bridge br-sdwan0 other-config:datapath-id=0000000000000001
```

Más adelante, en la **Figura 58** se configura el acceso a OVSDb mediante el puerto TCP 6632. Para ello se emplea el comando 'ovs-vsctl set-manager tcp:6632'. Luego, se verifica que el puerto se encuentre en estado de escucha con el comando 'netstat -tuln | grep 6632', donde se evidencia que el puerto está en estado LISTEN, lo que indica que el servicio OVSDb está disponible para futuras conexiones. Adicionalmente, se visualiza la configuración completa del puente 'br-sdwan0', donde se puede observar el puerto de conexión a OVSDb, el socket de conexión al controlador SDN y las interfaces que conforman este puente.

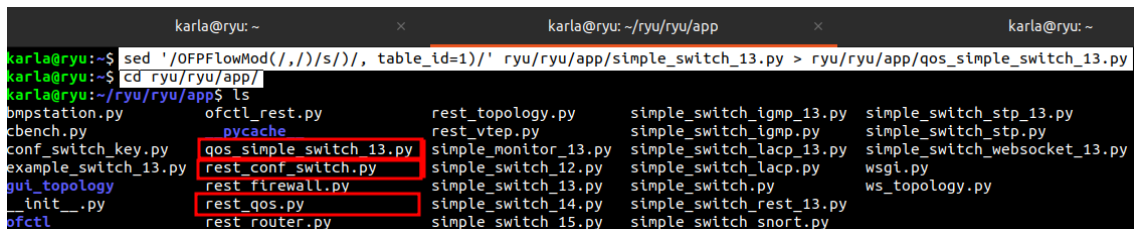
**Figura 58**  
Configuración de puerto para acceso a OVSDb.

```
root@OpenWrt:~# ovs-vsctl set-manager tcp:6632
root@OpenWrt:~# netstat -tuln | grep 6632
tcp        0      0 0.0.0.0:6632          0.0.0.0:*             LISTEN
root@OpenWrt:~# ovs-vsctl show
26d3001c-ce9e-45ba-b5ed-fb16b9ebd2f9
  Manager "tcp:6632"
  Bridge br-sdwan0
    Controller "tcp:192.168.15.10:6653"
    Port phy0-ap0
      Interface phy0-ap0
    Port br-lan
      Interface br-lan
    Port eth1
      Interface eth1
    Port br-sdwan0
      Interface br-sdwan0
        type: internal
  ovs_version: "2.17.9"
```

Desde el controlador, es necesario modificar el ID de las tablas que gestionan las entradas de flujos en el script ‘simple\_switch.py’. Esta es una acción necesaria para asegurar la interacción con ‘rest\_conf\_switch.py’, dado a que este último asume que la tabla a analizar posee el ID=1. Para realizar esta modificación se emplea el comando ‘sed’, el cual es un editor de flujo que permite buscar patrones específicos en el texto y reemplazarlos por otros (Ryu, 2014). Así, la sentencia empleada se visualiza en la **Figura 59**, donde se establece que el rango de líneas de búsqueda empezará en ‘OFPFLOWMod(’ y terminará en ‘)’’, en dicho rango se realizará la sustitución de ‘)’’ por el parámetro ‘, table\_id=1)’’, y estas modificaciones se guardaran en un nuevo script denominado ‘qos\_simple\_switch.py’. De este modo, se obtienen los scripts necesarios para la red propuesta.

### Figura 59

*Modificación de ID en tablas de entrada de flujo.*



```

karla@ryu: ~
┌───────────┴───────────┐
karla@ryu: ~$ sed '/OFPFLOWMod(/,/)s/)/, table_id=1)/' ryu/ryu/app/simple_switch_13.py > ryu/ryu/app/qos_simple_switch_13.py
karla@ryu: ~$ cd ryu/ryu/app/
karla@ryu: ~/ryu/ryu/app$ ls
bmpstation.py      ofctl_rest.py      rest_topology.py   simple_switch_igmp_13.py  simple_switch_stp_13.py
cbench.py          pycache            rest_vtep.py       simple_switch_igmp.py    simple_switch_stp.py
conf_switch_key.py qos_simple_switch_13.py  simple_monitor_13.py  simple_switch_lacp_13.py  simple_switch_websocket_13.py
example_switch_13.py rest_conf_switch.py    simple_switch_12.py   simple_switch_lacp.py    wsgi.py
gui_topology       rest_firewall.py      simple_switch_13.py   simple_switch.py         ws_topology.py
__init__.py        rest_qos.py           simple_switch_14.py   simple_switch_rest_13.py
ofctl              rest_router.py        simple_switch_15.py   simple_switch_snort.py
  
```

A continuación, es momento de ejecutar los scripts que proporcionarán funcionalidad a la red. Esto se realiza desde el controlador con el comando ryu-manager, donde se especifican los scripts ‘qos\_simple\_switch.py’, ‘rest\_conf\_switch.py’ y ‘rest\_qos.py’, como se muestra en la **Figura 60**. La salida de este comando brindará información sobre el estado de la red en funcionamiento, permitiendo visualizar que los puntos de acceso se conectan a la red usando los identificadores configurados al inicio de esta sección.

**Figura 60**

*Ejecución de programas desde controlador SDN.*

```

karla@ryu:~/ryu/ryu/app$ ryu-manager qos_simple_switch_13.py rest_conf_switch.py rest_qos.py
loading app qos_simple_switch_13.py
loading app rest_conf_switch.py
loading app rest_qos.py
loading app ryu.controller.ofp_handler
instantiating app None of ConfSwitchSet
creating context conf_switch
instantiating app None of DPSet
creating context dpset
creating context wsgi
instantiating app qos_simple_switch_13.py of SimpleSwitch13
instantiating app rest_conf_switch.py of ConfSwitchAPI
instantiating app rest_qos.py of RestQoSAPI
instantiating app ryu.controller.ofp_handler of OFPHandler
(8639) wsgi starting up on http://0.0.0.0:8080
[QoS][INFO] dpid=0000000000000001: Join qos switch.

```

Unión de AP1 a la red

Posterior a la conexión de los APs en la red, es necesario conectar las funciones del controlador SDN con OVSDb para permitir la gestión de QoS. Esto se realiza en el controlador mediante una solicitud PUT, en la cual se actualiza el valor de la variable 'ovsdb\_addr' con el socket de conexión a OVSDb, el cual contiene la dirección IP del AP y el puerto 6632, como se muestra en la **Figura 61**.

**Figura 61**

*Solicitud de conexión de controlador con OVSDb de API.*

```

karla@ryu: ~
karla@ryu: ~/ryu/ryu/app
karla@ryu: ~
ryu:~$ curl -X PUT -d '{"tcp:192.168.15.11:6632"}' http://localhost:8080/v1.0/conf/switches/0000000000000001/ovsdb_addr
ryu:~$

```

De este modo, en el terminal donde se están ejecutando los scripts que brindan funcionalidad en la red, es posible visualizar la respuesta a la solicitud realizada, indicando que fue aceptada correctamente y marcada con el código de estado 201, lo que confirma el éxito en la conexión, tal como se evidencia en la **Figura 62**.

**Figura 62**

*Respuesta exitosa a conexión con OVSDB.*

```

karla@ryu: ~
karla@ryu: ~/ryu/app
(8639) wsgi starting up on http://0.0.0.0:8080
[QoS][INFO] dpid=0000000000000001: Join qos switch.
packet in 0000000000000001 08:00:27:3d:54:b2 33:33:00:00:00:02 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 00:e0:81:37:09:65 5
packet in 0000000000000001 00:e0:81:37:09:65 b4:a9:fc:dc:11:ff 4294967294
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 00:e0:81:37:09:65 5
(8639) accepted ('127.0.0.1', 45176)
127.0.0.1 - - [07/Jul/2024 18:58:47] "PUT /v1.0/conf/switches/0000000000000001/ovsdb addr HTTP/1.1" 201 120 0.005092
packet in 0000000000000001 08:00:27:3d:54:b2 00:e0:81:37:09:65 5

```

### 3.3.2.2. Configuración de colas para cada AP

Para realizar adecuadamente la designación de AB en cada una de las colas, se procede a medir el AB de la red inalámbrica usando la herramienta iperf3. Para ello, se establece un entorno cliente-servidor con estaciones conectadas a esta red. El servidor se configura en una máquina Windows 10 y el cliente en un smartphone con sistema operativo Android 11. Desde el CMD de Windows se inicializa el servidor con el comando ‘iperf3.exe -s’ y, en el cliente se utiliza la aplicación “*Network tools*”, en la cual se ejecuta el comando ‘iperf3 -c 192.168.15.131 -t 30 -b 100M’, dicho comando especifica una prueba de 30 segundos con un ancho de banda de 100 Mbps, permitiendo evaluar la calidad de conexión entre el cliente y el servidor. Así, los resultados de esta prueba se muestran en la **Figura 63**, donde se determina que el ancho de banda máximo de la red es de 40 Mbps, valor que se utilizará para la gestión de colas en la red inalámbrica.

**Figura 63**  
*Velocidad máxima de red inalámbrica en servidor Iperf3.*

```

Server listening on 5201 (test #6)
-----
Accepted connection from 192.168.15.102, port 49698
[ 5] local 192.168.15.131 port 5201 connected to 192.168.15.102 port 49700
[ ID] Interval          Transfer          Bitrate
[ 5] 0.00-1.00 sec    4.25 MBytes      35.6 Mbits/sec
[ 5] 1.00-2.00 sec    5.00 MBytes      42.0 Mbits/sec
[ 5] 2.00-3.00 sec    4.62 MBytes      38.8 Mbits/sec
[ 5] 3.00-4.00 sec    4.75 MBytes      39.8 Mbits/sec
[ 5] 4.00-5.00 sec    5.12 MBytes      43.0 Mbits/sec
[ 5] 5.00-6.00 sec    4.50 MBytes      37.8 Mbits/sec
[ 5] 6.00-7.00 sec    4.88 MBytes      40.9 Mbits/sec
[ 5] 7.00-8.00 sec    4.88 MBytes      40.9 Mbits/sec
[ 5] 8.00-9.00 sec    4.38 MBytes      36.7 Mbits/sec
[ 5] 9.00-10.00 sec   5.12 MBytes      43.0 Mbits/sec
[ 5] 10.00-11.00 sec   4.62 MBytes      38.8 Mbits/sec
[ 5] 11.00-12.00 sec   4.62 MBytes      38.8 Mbits/sec
[ 5] 12.00-13.00 sec   5.00 MBytes      41.9 Mbits/sec
[ 5] 13.00-14.00 sec   4.38 MBytes      36.7 Mbits/sec
[ 5] 14.00-15.00 sec   5.00 MBytes      42.0 Mbits/sec
[ 5] 15.00-16.00 sec   5.00 MBytes      41.9 Mbits/sec
[ 5] 16.00-17.00 sec   4.50 MBytes      37.8 Mbits/sec
[ 5] 17.00-18.00 sec   4.88 MBytes      40.9 Mbits/sec
[ 5] 18.00-19.00 sec   4.88 MBytes      40.9 Mbits/sec
[ 5] 19.00-20.00 sec   4.25 MBytes      35.6 Mbits/sec
[ 5] 20.00-21.00 sec   5.00 MBytes      42.0 Mbits/sec
[ 5] 21.00-22.00 sec   4.75 MBytes      39.9 Mbits/sec
[ 5] 22.00-23.00 sec   4.62 MBytes      38.8 Mbits/sec
[ 5] 23.00-24.00 sec   5.00 MBytes      41.9 Mbits/sec
[ 5] 24.00-25.00 sec   4.38 MBytes      36.7 Mbits/sec
[ 5] 25.00-26.00 sec   5.00 MBytes      42.0 Mbits/sec
[ 5] 26.00-27.00 sec   5.00 MBytes      41.9 Mbits/sec
[ 5] 27.00-28.00 sec   4.50 MBytes      37.7 Mbits/sec
[ 5] 28.00-29.00 sec   5.00 MBytes      41.9 Mbits/sec
[ 5] 29.00-30.00 sec   4.88 MBytes      40.9 Mbits/sec
-----
[ ID] Interval          Transfer          Bitrate
[ 5] 0.00-30.01 sec    143 MBytes      39.9 Mbits/sec
receiver

```

En este sentido, considerando que lo recomendable es emplear el 70% del AB existente para asegurar una comunicación estable, incluso en situaciones de congestión. Se procede a calcular este porcentaje mediante una regla de tres simple, como se muestra en la **Ec. 4**, determinando que el AB máximo para la gestión de colas es de 28 Mbps.

$$AB_{70\%} = \left(\frac{70}{100}\right) \cdot (AB \text{ real}) \quad (\text{Ec. 4})$$

$$AB_{70\%} = \left(\frac{70}{100}\right) \cdot (40 \text{ Mbps})$$

$$AB_{70\%} = 28 \text{ Mbps}$$

Así, se establece la creación de cuatro colas en la interfaz inalámbrica de los APs en la red. Los rangos de cada cola se determinan en función de los porcentajes asignados en la planificación de QoS (ver **Tabla 12**). De esta forma, la cola con ID 0 tiene la menor prioridad y se asigna al tráfico BE. La cola con ID 1 se destina al tráfico FTP, la cola con ID 2 al tráfico

TCP proveniente de VOD, y la cola con ID 3 se reserva para el tráfico RTP y SIP. Los parámetros de cada cola se presentan en la **Tabla 17**.

**Tabla 17**

*Parámetros a considerar para configuración de colas.*

ID de cola	AB mínimo	AB máximo
0	1 Mbps	2 Mbps
1	1 Mbps	10 Mbps
2	1 Mbps	15 Mbps
3	1 Mbps	6 Mbps

Para crear las colas mencionadas se emplean solicitudes POST, donde se especifica la interfaz `pyh0-ap0`, que corresponde al nombre de la interfaz inalámbrica en OpenWrt, asimismo, se establecen los valores máximo y mínimo AB, y la URL del recurso correspondiente al identificador de cada AP, tal como se evidencia en la **Figura 64**. En esta figura, se observa que, tras enviar la solicitud, se recibe una respuesta en formato JSON que informa sobre el estado de la solicitud, en dicha información se visualiza el parámetro `'result: success'`, que indica que la configuración de QoS especificada en la solicitud se aplicó correctamente. Si surgiera algún problema, el campo `'result'` podría indicar `'failure'` y agrega detalles adicionales sobre el error. Además, en la **Figura 65** capturada en el terminal principal donde se ejecutan los programas de red, se evidencia la respuesta HTTP con el código de estado 200, confirmando de la misma forma el éxito en la solicitud.

**Figura 64**

*Configuración de colas mediante solicitudes POST.*

```

karla@ryu:~$
karla@ryu:~$ curl -X POST -d '{"port_name": "pyh0-ap0", "type": "linux-htb", "max_rate": "4000000", "queues":[{"max_rate": "3000000", "min_rate": "1000000"}, {"max_rate": "1000000", "min_rate": "3000000"}, {"max_rate": "1500000", "min_rate": "7000000"}, {"max_rate": "2800000", "min_rate": "1500000"}]}' http://localhost:8080/qos/queue/0000000000000001 -- Solicitud POST
[{"switch_id": "0000000000000001", "command_result": {"result": "success", "details": {"0": {"config": {"max-rate": "3000000", "min-rate": "1000000"}}, "1": {"config": {"max-rate": "1000000", "min-rate": "3000000"}}, "2": {"config": {"max-rate": "1500000", "min-rate": "7000000"}}, "3": {"config": {"max-rate": "2800000", "min-rate": "1500000"}}}}]}]karla@ryu:~$
karla@ryu:~$
karla@ryu:~$

```

↓  
Respuesta JSON

**Figura 65**

*Respuesta a solicitud POST en configuración de colas.*

```

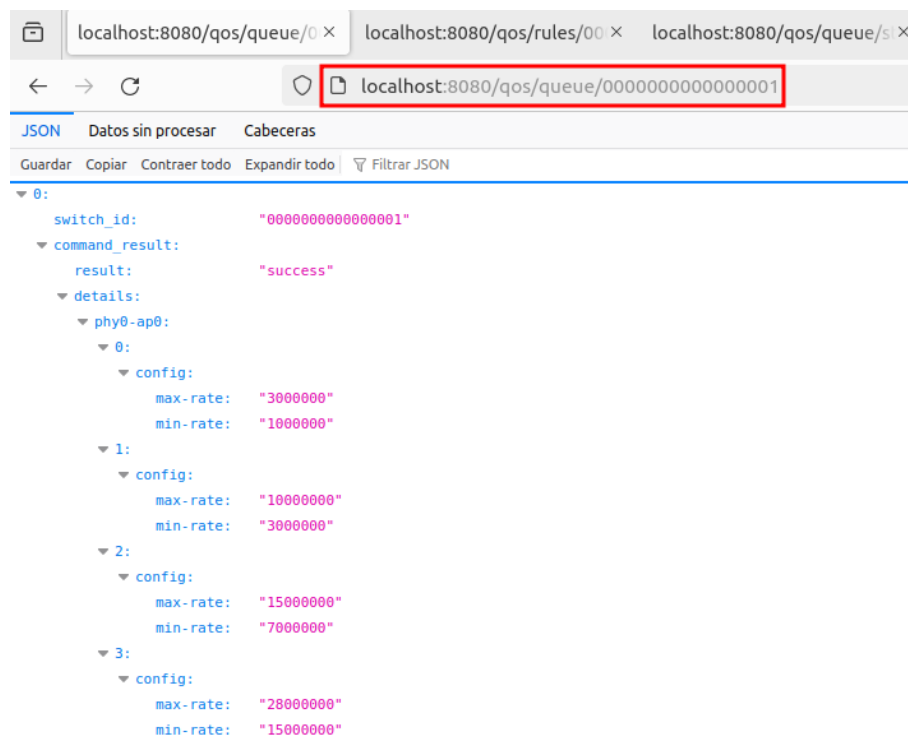
karla@ryu: ~/ryu/ryu/app
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
(10312) accepted ('127.0.0.1', 51358)
127.0.0.1 - - [09/Jul/2024 14:21:24] "POST /qos/queue/0000000000000001 HTTP/1.1" 200 397 0.225533
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:16 5

```

La **Figura 66** muestra la configuración realizada vista desde un navegador web. Para acceder a esta información, se debe introducir la URL del recurso a consultar en la barra de direcciones del navegador, lo que permitirá visualizar las configuraciones en formato JSON.

**Figura 66**

*Visualización de colas configuradas desde navegador.*



```

localhost:8080/qos/queue/0000000000000001
JSON  Datos sin procesar  Cabeceras
Guardar Copiar Contraer todo Expandir todo Filtar JSON
0:
  switch_id: "0000000000000001"
  command_result:
    result: "success"
  details:
    phy0-ap0:
      0:
        config:
          max-rate: "3000000"
          min-rate: "1000000"
      1:
        config:
          max-rate: "10000000"
          min-rate: "3000000"
      2:
        config:
          max-rate: "15000000"
          min-rate: "7000000"
      3:
        config:
          max-rate: "28000000"
          min-rate: "15000000"

```

### 3.3.2.3. Parametrización de marcaje en la red

Para la configuración de marcaje en la red, la API REST recibe los siguientes parámetros; dirección IP destino, puerto de comunicación, protocolos de red (UDP, TCP e ICMP). Además, se debe proporcionar el valor decimal de DSCP que se aplicará a los paquetes que coincidan

con estos parámetros. En este sentido, la **Tabla 18** muestra los parámetros que se empleará para la configuración de marcaje en la red.

**Tabla 18**  
*Parámetros de marcaje en la red.*

Dirección IP destino	Puerto/s	Protocolo	DSCP
192.168.15.232	16384 - 32767	UDP	46 (EF)
192.168.15.232	5060	UDP	24 (CS3)
192.168.15.231	8096	TCP	28 (AF32)
192.168.15.231	21	TCP	22 (AF23)

Mediante el uso de solicitudes POST, se lleva a cabo la configuración de marcaje en la red, tal como se muestra en la **Figura 67**. Además, al revisar las respuestas JSON de cada solicitud, es posible observar que todas han tenido éxito en la implementación. De manera similar, la **Figura 68**, indica que en todas las solicitudes se obtuvo una respuesta HTTP marcadas con el código de estado 200.

**Figura 67**  
*Configuración de marcaje mediante solicitudes POST.*

```

karla@ryu:~$ curl -X POST -d '{"match": {"nw_dst": "192.168.15.232", "nw_proto": "UDP", "tp_dst": "16384"}, "actions": {"mark": "46"}}' http://localhost:8080/qos/rules/0000000000000001
{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=5"}]}
karla@ryu:~$ curl -X POST -d '{"match": {"nw_dst": "192.168.15.232", "nw_proto": "UDP", "tp_dst": "5060"}, "actions": {"mark": "24"}}' http://localhost:8080/qos/rules/0000000000000001
{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=6"}]}
karla@ryu:~$ curl -X POST -d '{"match": {"nw_dst": "192.168.15.231", "nw_proto": "TCP", "tp_dst": "8096"}, "actions": {"mark": "28"}}' http://localhost:8080/qos/rules/0000000000000001
{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=7"}]}
karla@ryu:~$ curl -X POST -d '{"match": {"nw_dst": "192.168.15.231", "nw_proto": "TCP", "tp_dst": "21"}, "actions": {"mark": "22"}}' http://localhost:8080/qos/rules/0000000000000001
{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=8"}]}
karla@ryu:~$

```

**Figura 68**  
*Respuestas a solicitudes POST de marcaje.*

```

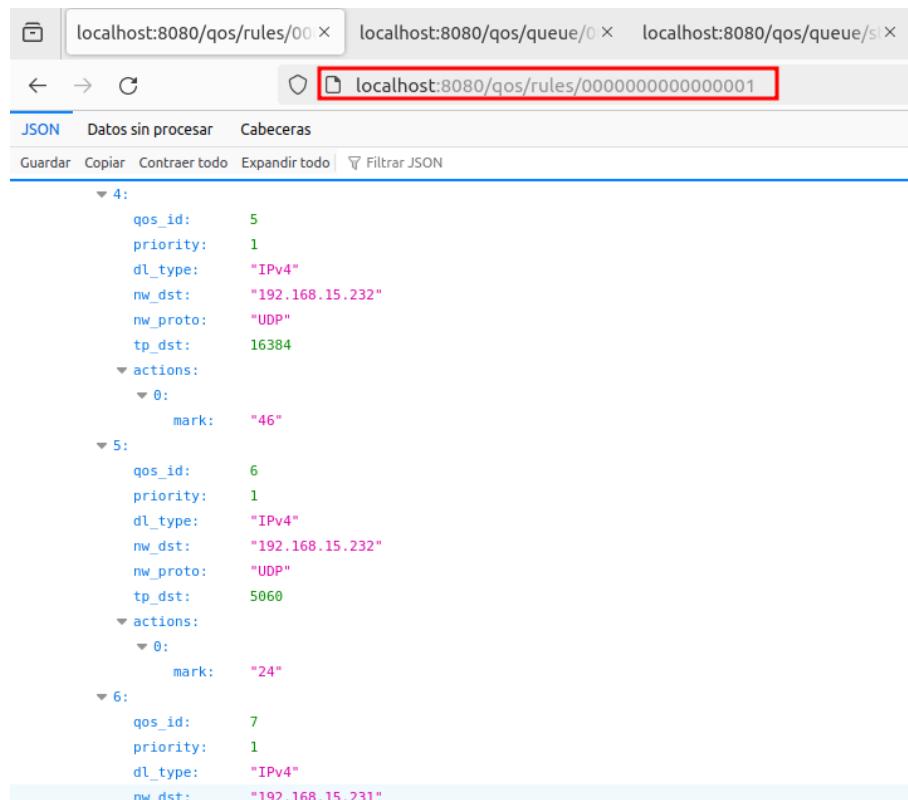
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
(10312) accepted ('127.0.0.1', 51194)
127.0.0.1 - - [09/Jul/2024 14:36:07] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 223 0.001865
(10312) accepted ('127.0.0.1', 35986)
127.0.0.1 - - [09/Jul/2024 14:36:19] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 224 0.002685
(10312) accepted ('127.0.0.1', 35690)
127.0.0.1 - - [09/Jul/2024 14:36:31] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 224 0.001676
(10312) accepted ('127.0.0.1', 50368)
127.0.0.1 - - [09/Jul/2024 14:36:46] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 224 0.001084
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5

```



Desde el navegador se comprueba las configuraciones realizadas, en este caso se visualiza las reglas establecidas en el recurso `http://localhost:8080/qos/rules/switch-id`, como se evidencia en la **Figura 69**.

**Figura 69**  
*Visualización de marcaje desde navegador.*



#### 3.3.2.4. Clasificación de tráfico

La clasificación de tráfico toma como referencia el campo DSCP de los paquetes para asignarlos a las colas correspondientes. En este caso, los paquetes BE se clasifican en la cola con ID 0, los paquetes marcados con AF23 se asignan a la cola con ID 1, los paquetes AF32 se clasifican en la cola con ID 2 y los paquetes EF y CS3 se transmiten a través de la cola con ID 3, como se muestra en la **Tabla 19**.

**Tabla 19**  
Clasificación de tráfico en las colas creadas.

PHB	Cola asignada
BE	0
AF23	1
AF32	2
EF, CS3	3

Las solicitudes POST utilizadas para esta configuración se evidencia en la **Figura 70** y las respuestas JSON indican que han sido aplicadas correctamente, hecho que también es visible en la **Figura 71**, ya que las respuestas HTTP de todas estas solicitudes se encuentran con el código de estado 200.

**Figura 70**  
Clasificación de tráfico mediante solicitudes POST.

```

karla@ryu:~$
karla@ryu:~$ curl -X POST -d '{"match": {"ip_dscp": "48"}, "actions": {"queue": "3"}}' http://localhost:8080/qos/rules/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=1"}]}karla@ryu:~$
karla@ryu:~$
karla@ryu:~$ curl -X POST -d '{"match": {"ip_dscp": "24"}, "actions": {"queue": "3"}}' http://localhost:8080/qos/rules/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=2"}]}karla@ryu:~$
karla@ryu:~$
karla@ryu:~$ curl -X POST -d '{"match": {"ip_dscp": "28"}, "actions": {"queue": "2"}}' http://localhost:8080/qos/rules/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=3"}]}karla@ryu:~$
karla@ryu:~$
karla@ryu:~$ curl -X POST -d '{"match": {"ip_dscp": "22"}, "actions": {"queue": "1"}}' http://localhost:8080/qos/rules/0000000000000001
[{"switch_id": "0000000000000001", "command_result": [{"result": "success", "details": "QoS added. : qos_id=4"}]}karla@ryu:~$
karla@ryu:~$

```

**Figura 71**  
Respuesta a solicitudes de clasificación de tráfico.

```

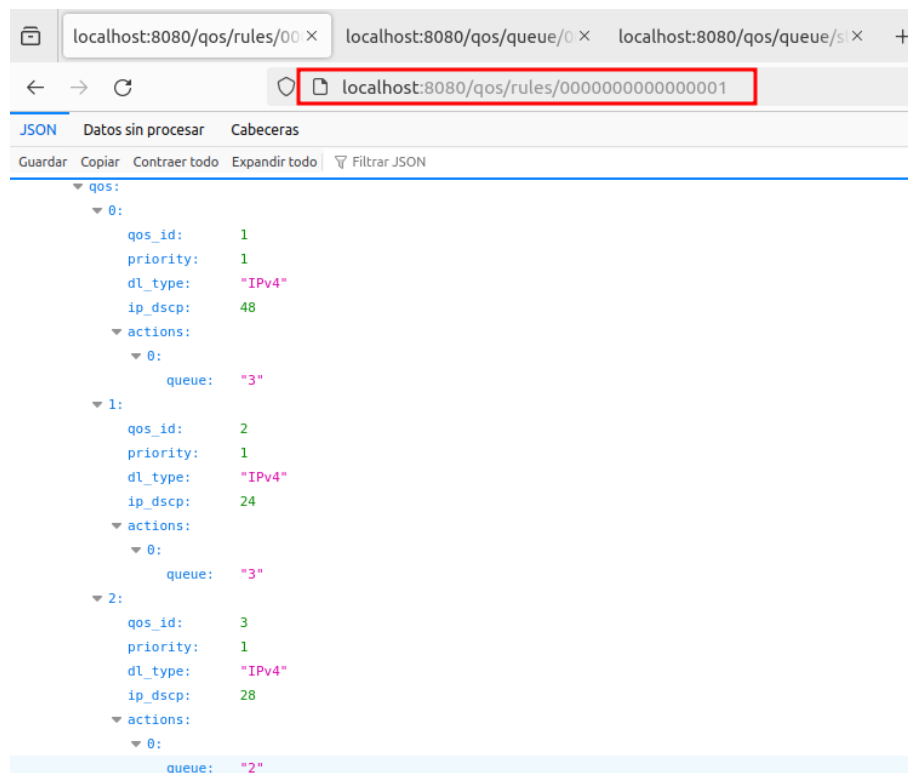
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
(10312) accepted ('127.0.0.1', 55866)
127.0.0.1 - - [09/Jul/2024 14:33:37] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 223 0.001202
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 33:33:00:00:00:fb 5
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
(10312) accepted ('127.0.0.1', 41216)
127.0.0.1 - - [09/Jul/2024 14:33:48] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 223 0.008916
packet in 0000000000000001 b4:a9:fc:dc:11:ff 01:00:5e:7f:ff:fa 5
(10312) accepted ('127.0.0.1', 51576)
127.0.0.1 - - [09/Jul/2024 14:33:57] "POST /qos/rules/0000000000000001 HTTP/1.1" 200 223 0.000904

```

Finalmente, se comprueba la clasificación de tráfico establecida. Considerando que, al igual que las reglas de marcaje, la clasificación de tráfico se establece en el punto final `http://localhost:8080/qos/rules/switch-id`. Por tanto, se introduce dicho punto final en el navegador y se visualiza las configuraciones realizadas, como se muestra en la **Figura 72**.

**Figura 72**

*Visualización de clasificación de tráfico desde navegador.*



### 3.3.2.5. Visualización de marcaje con Wireshark

Este apartado aborda la verificación de las políticas de QoS configuradas en la red SDWN, con un enfoque particular en el marcaje de paquetes, debido a que este es un proceso necesario para distinguir y clasificar el tráfico de red de manera efectiva. En este sentido, la validación de marcaje se realiza mediante la herramienta Wireshark, misma que permite capturar y analizar tráfico de red.

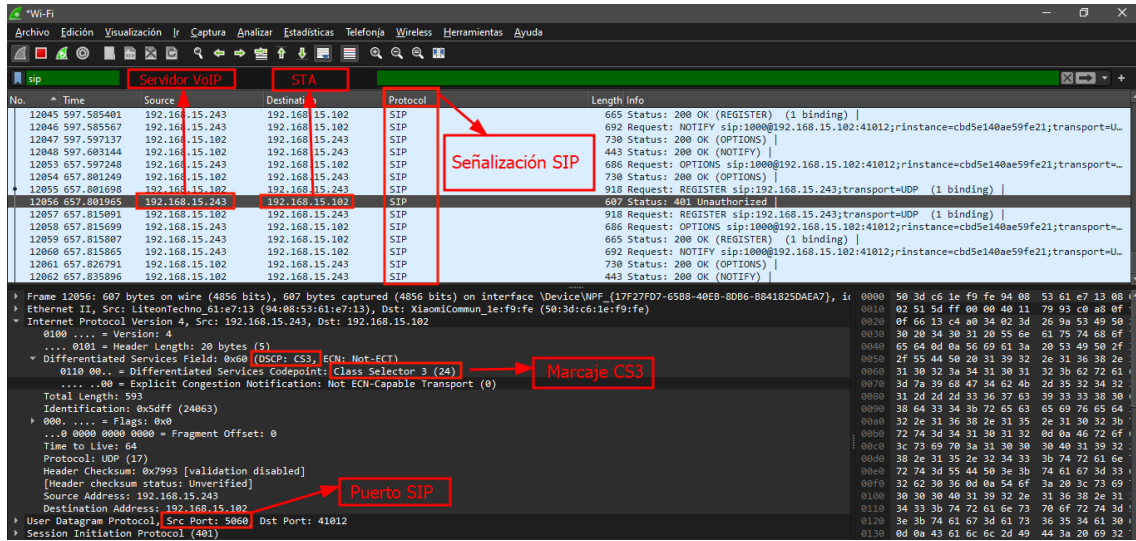
Para capturar el tráfico, se debe ejecutar Wireshark en una máquina conectada de manera inalámbrica a la red SDWLAN. Una vez abierto Wireshark, se selecciona la interfaz de red inalámbrica correspondiente y se procede a iniciar la captura de paquetes. A continuación, en la red se ejecutan los servidores de VoIP, VOD y FTP, y desde los clientes inalámbricos se genera tráfico hacia estos servidores. Este proceso genera un flujo de tráfico real, que permite analizar el marcaje del tráfico en la red y verificar la implementación de Calidad de Servicio (QoS).

Al finalizar la captura de tráfico, se inspeccionan los paquetes generados por los servidores para verificar el valor del campo DSCP. Este campo se puede localizar en la sección "Internet Protocol Version 4" dentro del panel de detalles de Wireshark. Para facilitar la localización de los paquetes transmitidos por los distintos servicios, se pueden aplicar filtros basados en protocolos o puertos específicos, lo que hace que la búsqueda sea más eficaz.

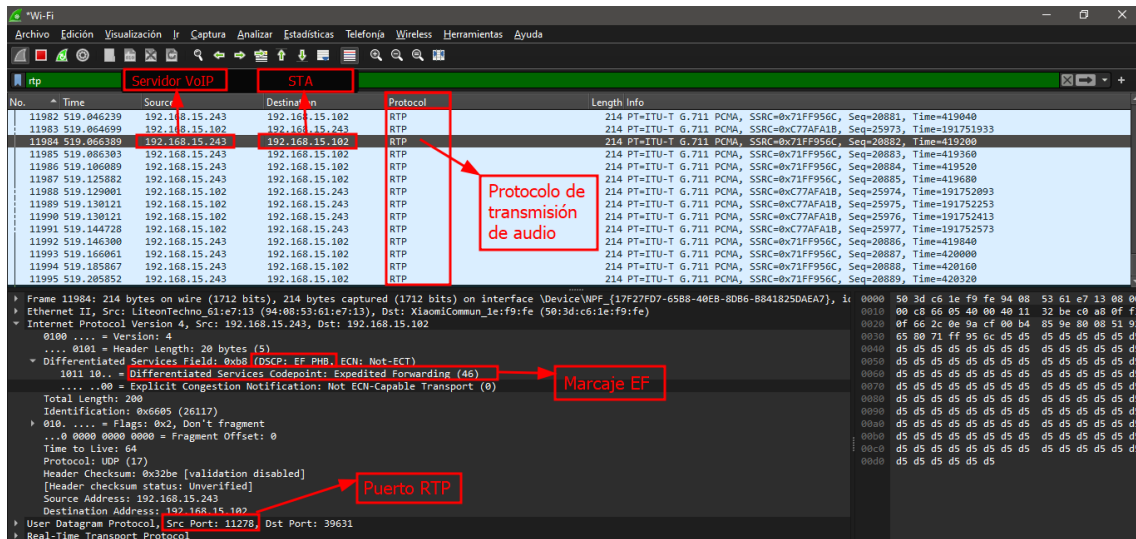
Comenzamos analizando el marcaje en paquetes referentes al servicio VoIP. Para ello, filtramos el tráfico del protocolo SIP, dado que es un protocolo fundamental para la señalización y el establecimiento de llamadas en VoIP. La **Figura 73** muestra la interacción cliente-servidor mediante el protocolo SIP, mismo que opera en el puerto 5060/UDP. Además, se puede observar que el campo DSCP está marcado como Class Selector 3 (CS3), lo que evidencia la correcta aplicación de las reglas configuradas.

De manera similar, se analiza el tráfico correspondiente al protocolo RTP, el cual se encarga del transporte de audio y video en tiempo real dentro del servicio VoIP. La **Figura 74** muestra los paquetes RTP, mismos que operan en el rango de puertos UDP comprendido entre 16384 y 32767. Asimismo, se verifica que el campo DSCP está marcado como Expedited Forwarding (EF), lo que asegura un envío rápido y una alta priorización para este tipo de paquetes.

**Figura 73**  
*Marcaje en paquetes SIP.*

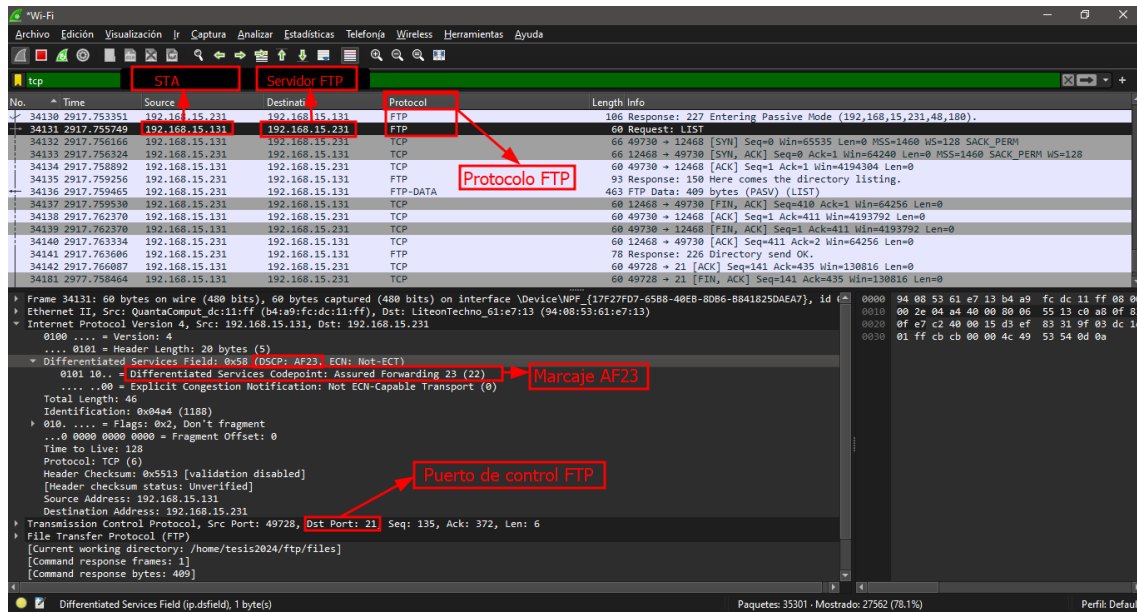


**Figura 74**  
*Marcaje en paquetes RTP.*



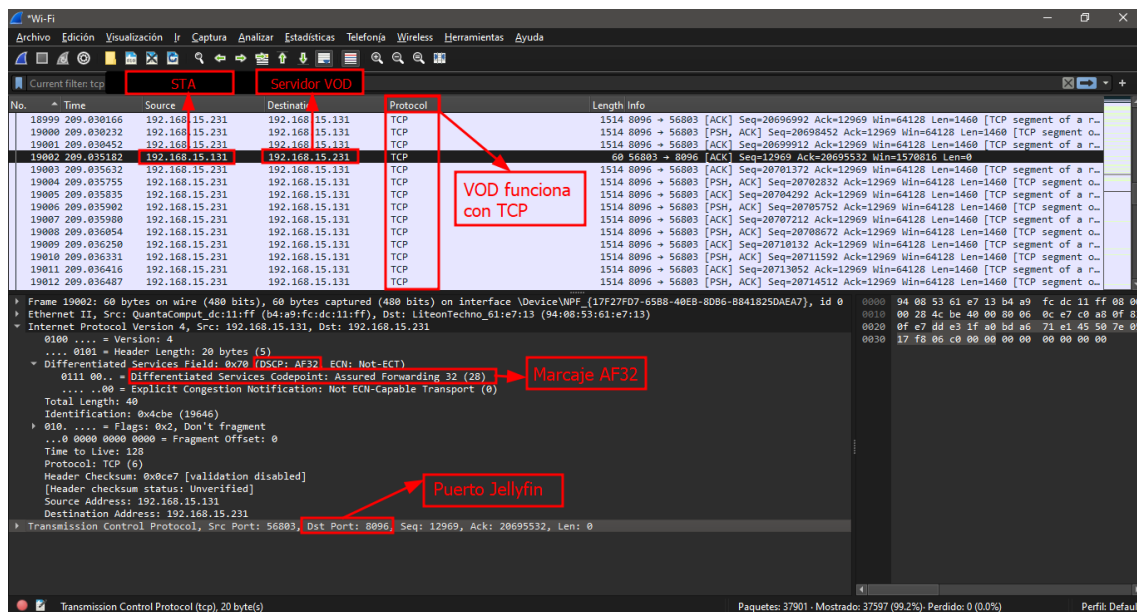
Continuamos analizando el tráfico referente al servicio FTP. La **Figura 75** muestra la interacción de cliente-servidor usando el protocolo FTP, en el que se emplea el puerto 21/TCP. En esta captura se puede observar que el campo DSCP se encuentra marcado como Assured Forwarding 23 (AF23), lo que confirma que el marcaje se ha aplicado correctamente para este servicio.

**Figura 75**  
*Marcaje en paquetes FTP.*



Finalmente, se analiza el tráfico generado por el servicio VOD. Dado a que este funciona sobre TCP, se usa este protocolo como filtro para localizar los paquetes correspondientes. La **Figura 76** muestra que los paquetes TCP que operan en el puerto 8096 (puerto Jellyfin) se encuentran marcados como Assured Forwarding (AF23) en el campo DSCP, indicando que en este servicio también se aplicó adecuadamente las políticas de QoS configuradas.

**Figura 76**  
*Marcaje en paquetes TCP debido al servicio de VOD.*



#### 4. CAPÍTULO IV: Pruebas y análisis de funcionamiento

Este capítulo aborda la fase de verificación del esquema planteado en la **Figura 16**. Así, en función de la QoS implementada, se analiza diversos parámetros de rendimiento de la red, tales como el Ancho de Banda (AB), la Tasa de Pérdida de Paquetes (IPLR), el Retardo de Transferencia de Paquetes (IPTD), el Tiempo de ida y vuelta (RTT) y la Disponibilidad de la red. Para llevar a cabo este análisis, se toma como referencia la recomendación ITU-T Y.1540, la cual establece directrices para realizar mediciones que permitan conocer la tendencia de estos parámetros bajo condiciones definidas.

##### 4.1.Planteamiento de pruebas conforme a ITU-T Y.1540

La **sección 2.2.3.2** muestra una explicación detallada de la recomendación ITU-T Y.1540. En el presente apartado se realiza la aplicación de dicha normativa para diseñar escenarios de pruebas que permitan verificar las configuraciones de QoS realizadas. Para ello, la normativa sugiere una serie de pasos, los cuales se describen a continuación y se desarrollan en las siguientes secciones.

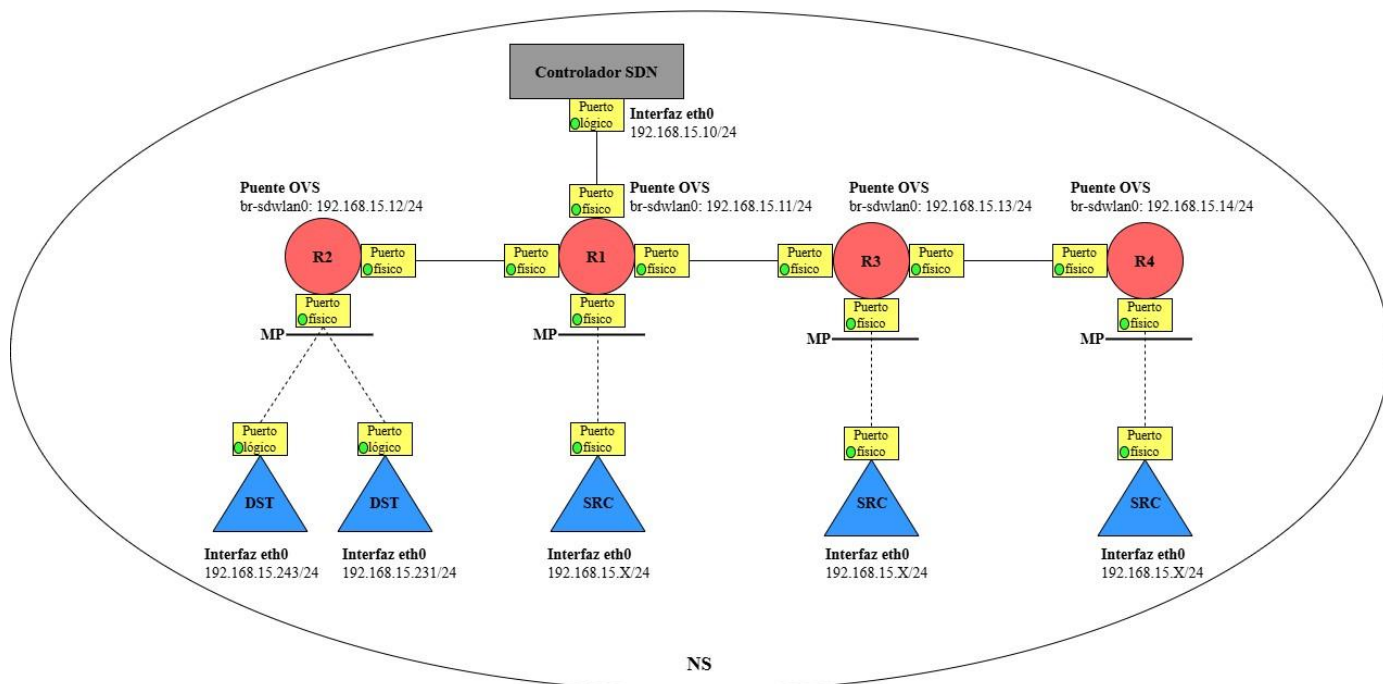
- **Paso 1:** Definir el esquema de red para las pruebas. Esto implica identificar las interfaces físicas y lógicas de la red, así como los componentes clave que participan en ella. Además, es necesario determinar los puntos de medición y los dispositivos que actuarán como origen y destino durante las pruebas.
- **Paso 2:** Establecimiento de un conjunto de KPIs en función de una determinada población de interés, así como las herramientas que se empleará para medirlos.
- **Paso 3:** Definir escenarios específicos para medir los KPIs establecidos. Dichos escenarios deben simular condiciones relevantes para medir el rendimiento de la red.
- **Paso 4:** Presentar un resumen de los resultados obtenidos en las mediciones. Este paso incluye un análisis detallado de los datos recopilados en función de los KPIs definidos.

#### 4.1.1. Topología de pruebas

De conformidad con el paso uno, en la **Figura 77** se define la topología de red que se empleará para las pruebas. Esta representación permite visualizar la configuración de las interfaces y la ubicación de los puntos de medición. A continuación, se describen los principales componentes y sus respectivas funciones.

- **Encaminador (R):** Dispositivo encargado de gestionar la trasmisión del tráfico de red.
- **Punto de medición (MP):** Punto de medición, donde se realiza el control de tráfico y se aplican sistemas de medición basado en software.
- **Origen del tráfico (SRC):** El origen del tráfico lo constituye las estaciones conectadas a la red inalámbrica.
- **Destino del tráfico (DST):** El destino del tráfico son los servidores de la red.
- **Sección de red (NS):** La constituye toda la topología de red (ver **Figura 77**).

**Figura 77**  
Esquema de pruebas conforme a ITU-T Y.1540.





#### 4.1.2. *Parámetros de rendimiento en transferencia de paquetes IP*

En cumplimiento del segundo paso, se define cinco KPIs para evaluar la QoS en la red SDWN. Estos parámetros son: Ancho de Banda (AB), Tasa de Pérdida de Paquetes (IPLR), Retardo de Transferencia de Paquetes (IPTD), Tiempo de ida y vuelta (RTT) y Disponibilidad de red. La selección de estos KPIs responde a la necesidad de cubrir los aspectos de velocidad, exactitud y seguridad de funcionamiento, tal como lo establece la recomendación ITU-T Y.1540. Además, esto permite delimitar el estudio y realizar un análisis detallado del rendimiento de la red con las configuraciones de QoS establecidas.

Para medir los KPIs, se han seleccionado dos herramientas de código abierto ampliamente utilizadas en la evaluación de redes; iPerf3 y hping3. Estas herramientas permiten generar y enviar paquetes TCP, UDP e ICMP, con la flexibilidad de personalizar sus características según los requisitos específicos de cada prueba (OffSec, 2024). De este modo, es posible realizar pruebas adecuadas que proporcionen una visión clara del comportamiento de la red bajo diferentes configuraciones y condiciones de tráfico. La **Tabla 20** muestra los KPIs que se medirán con cada herramienta.

Adicionalmente, en este estudio se emplea la herramienta Wireshark para visualizar el marcaje de paquetes. Aunque esta característica no forma parte de los parámetros de rendimiento de la red, es fundamental para verificar la correcta aplicación de QoS.

**Tabla 20**

*Herramientas de medición seleccionadas.*

Herramienta	KPIs
iPerf3	Ancho de banda (AB) Disponibilidad de red Tasa de pérdida de paquetes (IPLR)
Hping3	Tiempo de ida y vuelta (RTT) Retardo de transferencia de paquetes (IPTD)

*Nota.* Cabe mencionar que el parámetro de disponibilidad no es una medición directa, sino que puede determinarse a partir del valor del parámetro IPLR.

#### 4.1.3. Configuración de pruebas para evaluación de QoS

De acuerdo con lo establecido en el tercer paso, en esta sección se presenta el despliegue de cuatro escenarios destinados a evaluar el rendimiento de la red. Estos escenarios se desarrollan considerando los KPIs, las herramientas de medición, una población de interés (densidad de tráfico) y un período de medición de 10 minutos. Es importante destacar que cada escenario es evaluado tanto antes como después de aplicar QoS en la red, con el objetivo de analizar la tendencia de estos parámetros en un entorno SDWN.

Para estos escenarios, es importante considerar que la red inalámbrica opera bajo el estándar 802.11n en la banda de 2,4 GHz, y la configuración de canales en los puntos de acceso (AP) se realizó de manera que no se solapen entre sí (ver **Tabla 11**). Finalmente, de acuerdo con la medición de velocidad realizada en la **sección 3.3.2.2**, se determinó que la velocidad máxima del enlace inalámbrico es de 40 Mbps, de manera que las pruebas de rendimiento reflejaran valores cercanos a este.

Por otro lado, el entorno de red SDWN cuenta con servidores de VOD, VoIP y FTP, que serán utilizados durante las pruebas (la **Tabla 21** muestra la dirección IP y puerto de cada servidor que se empleará en las pruebas). Así, para cada escenario se genera tráfico TCP y UDP con diferente densidad de tráfico, lo cual permitirá analizar el impacto de las configuraciones de QoS aplicadas. La **Tabla 22** presenta los escenarios planteados, y en las secciones siguientes se detalla la ejecución de cada uno de ellos.

**Tabla 21**

*Dirección IP y puerto de servicios para pruebas.*

<b>Servidor</b>	<b>Dirección IP</b>	<b>Puerto</b>
VoIP	192.168.15.243	16384/UDP
VOD	192.168.15.231	8096/TCP
FTP	192.168.15.231	21/TCP

**Tabla 22***Descripción de escenarios para evaluación de red.*

Escenarios	Tamaño de paquetes	Cantidad de clientes	Tiempo de medición
Escenario 1	1500 bytes	3 clientes (un cliente por cada AP)	10 minutos por cada escenario
Escenario 2	30000 bytes		
Escenario 3	45000 bytes		
Escenario 4	65507 bytes		

#### 4.1.3.1. Escenarios evaluados antes de aplicar QoS

Este apartado presenta la evaluación de los cuatro escenarios en la red sin configuraciones de QoS, con el objetivo de obtener un punto de referencia sobre el comportamiento de los cinco KPIs definidos. Además, se busca identificar las limitaciones de la red bajo condiciones normales, donde todos los flujos de datos compiten por los recursos de red. Los datos recopilados durante estas evaluaciones permitirán comparar el rendimiento de la red antes y después de la aplicación de QoS. Tanto los datos obtenidos como los scripts utilizados para las gráficas están disponibles en el repositorio de GitHub: <https://github.com/knmoncayo/pruebas-sdwan-tesis-utn.git>.

##### 4.1.3.1.1. Escenario 1: Tráfico ligero

Para comenzar con la evaluación del primer escenario, se lleva a cabo la preparación de los servidores y clientes de la red, garantizando la conectividad entre ellos, y el correcto funcionamiento de las herramientas de medición. A continuación, se evalúan y recopilan datos sobre los cinco KPIs establecidos en función de la población de interés, que en este caso consiste en paquetes de 1500 bytes. Finalmente, se presenta una gráfica que muestra la tendencia de los datos recopilados durante el periodo medido.

- **Ancho de Banda (AB)**

Para medir el ancho de banda, se inicializa el servidor iPerf3 en las máquinas virtuales que alojan los servidores VoIP, VOD y FTP. Para esto se emplea el comando `iperf3 -s -p`

<puerto>, donde el parámetro <puerto> especifica el puerto utilizado por cada servidor. En este caso, se emplean los puertos 16384/UDP para VoIP, 8096/TCP para VOD y 21/TCP para FTP.

Los clientes generan tráfico hacia un servidor iPerf3 específico utilizando el comando `iperf3 -c <direccion-ip-servidor> -p <puerto> -l 1500 -t 600`. El parámetro `-l 1500` define el tamaño de paquetes en bytes, mientras que, el parámetro `-t 600` indica la duración de la prueba en segundos. Y para el servidor iPerf3 que funciona bajo el puerto 16384/UDP, el cliente correspondiente debe agregar el parámetro `-u` al comando anterior, para indicar el uso del protocolo UDP. Así, el comando completo sería `iperf3 -c <direccion-ip-servidor> -p <puerto> -l 1500 -t 600 -u`.

Luego de efectuar la prueba, se obtiene un resumen detallado, donde muestra, segundo a segundo, la variación de ancho de banda en cada servidor iPerf3. De este modo, se emplea la herramienta Excel para tratar y ordenar los datos obtenidos, para posteriormente graficarlos. El tratamiento de datos se lleva a cabo filtrando los paquetes en intervalos de cinco segundos para facilitar su interpretación, además el valor de AB se expresa en Mbps.

El gráfico de los datos mencionados se realiza utilizando el lenguaje de programación Python3 junto con las librerías `matplotlib`<sup>28</sup> y `pandas`<sup>29</sup>, resultando en la **Figura 78**. En esta figura se observa que el servicio VOD presenta un mayor ancho de banda durante la mayor parte de la prueba, alcanzando un valor máximo de 32 Mbps. El servicio FTP mantiene un ancho de banda bajo, pero hay instantes en donde el AB de este servicio incrementa, alcanzado un valor máximo de 44 Mbps. Finalmente, el servicio VoIP presenta el menor AB durante la prueba, con 1,5 Mbps como valor máximo.

---

<sup>28</sup> `matplotlib` es una librería de Python que permite crear gráficos estadísticos, ya sean estáticos, animados o visualizaciones interactivas (`matplotlib-org`, 2024).

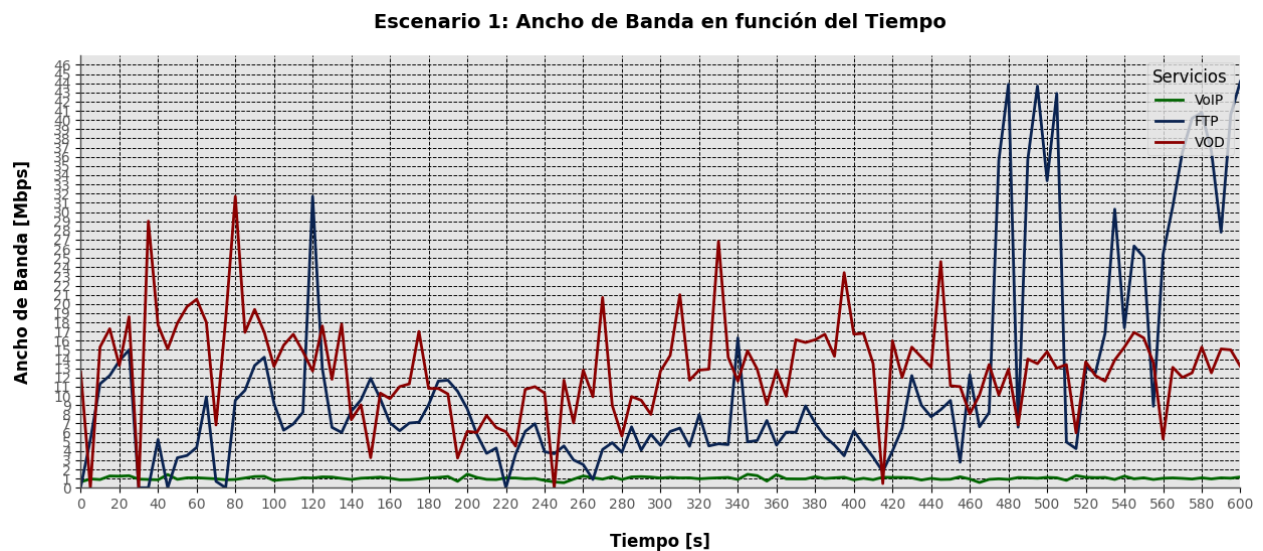
<sup>29</sup> `pandas` es una librería de Python que permite trabajar con conjuntos de datos, ofrece las opciones de analizar, limpiar, explorar y manipular datos (`W3schools`, 2024).

Dado que en este escenario se trabaja con un MTU de 1500 bytes, un valor de referencia en redes de datos, se observa que los servicios VOD y FTP fluctúan entre valores altos, alcanzando el ancho de banda máximo de la red, es decir, 40 Mbps. Esto ocurre porque dichos servicios demandan una mayor cantidad de recursos de red y, al ser una red sin QoS, estos servicios compiten constantemente por los recursos disponibles (Laassiri et al., 2017).

Además, los bajos valores de ancho de banda presentados por el servicio VoIP se deben a que el protocolo UDP, utilizado para VoIP, requiere menos ancho de banda debido a la compresión y al envío constante de datos en paquetes pequeños. Tal característica ayuda a reducir la latencia y el impacto de la fragmentación de paquetes (Vasco, 2010). Por lo tanto, aunque VoIP no necesita mucho ancho de banda, sí requiere estabilidad y baja latencia, lo cual se puede lograr con QoS.

### Figura 78

*Evaluación de Ancho de banda en escenario 1, sin QoS.*



- **Tasa de paquetes perdidos (IPLR)**

Los datos sobre la cantidad de paquetes perdidos durante la prueba se recopilaron utilizando iperf3 para el caso de la conexión UDP, y la herramienta Wireshark para el protocolo TCP. Los comandos de iperf3 empleados fueron los mismos de la sección anterior, mientras que para

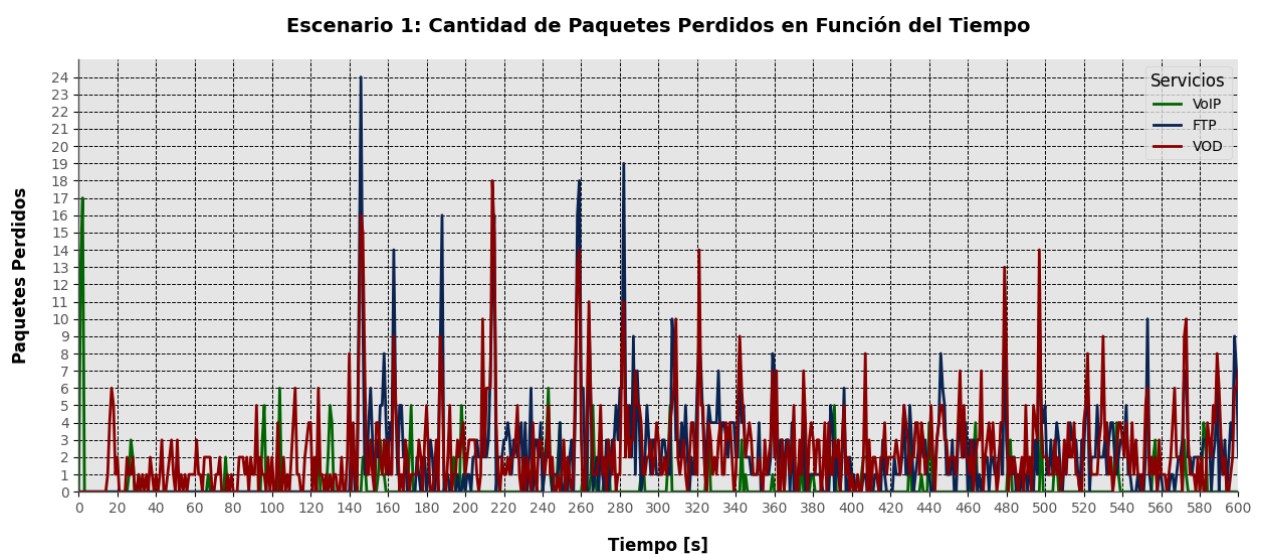
Wireshark se empleó los filtros; `tcp.port==21 && tcp.analysis.lost_segment` y `tcp.port==8096 && tcp.analysis.lost_segment`, para conocer la cantidad de paquetes perdidos en los servicios VOD y FTP, respectivamente.

La **Figura 79** muestra la gráfica de paquetes perdidos en función del tiempo, el intervalo de tiempo seleccionado es de un segundo, permitiendo la representación de todos los paquetes perdidos de cada servicio. Además, es posible visualizar que el servicio con pérdidas más altas es VOD, a esto le sigue el servicio FTP, y finalmente el servicio VoIP.

De acuerdo con Cranley & Davis (2005), el valor de MTU impacta directamente en los requerimientos de la red para transmitir datos. En este caso, al utilizar un MTU de 1500 bytes, la pérdida de paquetes no es significativa lo que permite el funcionamiento adecuado de los servicios en la red. Además, se observa instantes donde los servicios presentan cero paquetes perdidos. Sin embargo, se espera que en la evaluación de los siguientes escenarios haya pérdidas más significativas debido a la fragmentación de paquetes, dado al uso de valores altos de MTU.

### Figura 79

*Evaluación de paquetes perdidos en escenario 1, sin QoS.*



Las herramientas iPerf3 y Wireshark permiten analizar la cantidad de paquetes enviados y perdidos, como se muestra en las **Figuras 75, 76 y 77**. A partir de estos datos, se calcula la Tasa de Pérdidas de Paquetes IP (IPLR), que representa la relación entre los paquetes perdidos y el total de paquetes enviados. De esta manera, en la **Tabla 23** es posible visualizar el IPLR de cada servicio, destacando que todos presentan valores semejantes.

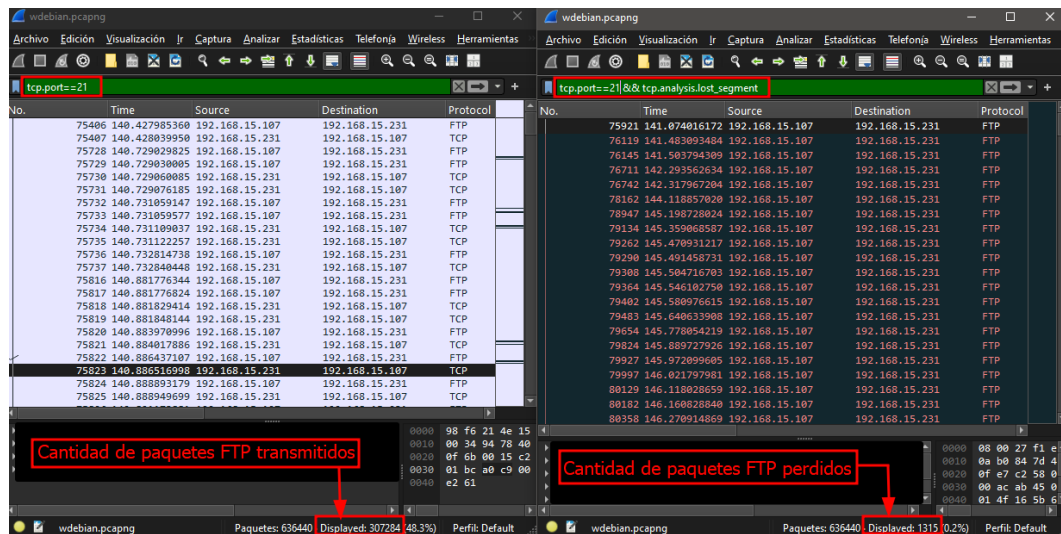
**Figura 80**  
Cantidad de paquetes perdidos y enviados para servicio VoIP.

```
[ 5] 596.00-597.00 sec 122 KBytes 996 Kbits/sec 7.977 ms 0/83 (0%)
[ 5] 597.00-598.00 sec 146 KBytes 1.20 Mbits/sec 6.322 ms 0/100 (0%)
[ 5] 598.00-599.00 sec 207 KBytes 1.69 Mbits/sec 4.363 ms 0/141 (0%)
[ 5] 599.00-600.00 sec 125 KBytes 1.02 Mbits/sec 3.847 ms 0/85 (0%)
[ 5] 600.00-600.14 sec 20.5 KBytes 1.17 Mbits/sec 3.858 ms 0/14 (0%)
-----
[ ID] Interval          Transfer      Bandwidth      Jitter          Lost/Total Datagrams
[ 5] 0.00-600.14 sec 0.00 Bytes 0.00 bits/sec 3.858 ms      223/52429 (0.43%)
-----
Server listening on 16384
-----
```

**Figura 81**  
Cantidad de paquetes perdidos y enviados para servicio VOD.

The figure consists of two side-by-side screenshots of the Wireshark network protocol analyzer. The left screenshot shows a packet list for the filter 'tcp.port==8096', displaying columns for No., Time, Source, Destination, and Protocol. The status bar at the bottom indicates 'Paquetes: 636440' and 'Disolvidos: 326256 (51.3%)'. A red box highlights this status bar with the text 'Cantidad de paquetes VOD transmitidos'. The right screenshot shows a packet list for the filter 'tcp.port==8096 && tcp.analysis.lost\_segment', displaying columns for No., Time, Source, Destination, and Protocol. The status bar at the bottom indicates 'Paquetes: 636440' and 'Disolvidos: 1409 (0.2%)'. A red box highlights this status bar with the text 'Cantidad de paquetes VOD perdidos'.

**Figura 82**  
Cantidad de paquetes perdidos y enviados para servicio FTP.



**Tabla 23**  
Datos de IPLR obtenidos en el Escenario 1, sin QoS

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	223	52429	0,00425
VOD	1409	326256	0,00431
FTP	1315	307284	0,00427

- **Disponibilidad de red**

De acuerdo con la normativa ITU-T Y.1540, la disponibilidad es un valor que se obtiene a partir del indicador IPLR. Durante un período de observación, la red se considera disponible si  $IPLR < c_1$ , siendo  $c_1$  un valor de referencia de 0,20, de lo contrario se considera no disponible (ITU-T Y.1540, 2019, p. 29)<sup>30</sup>. Entonces, conociendo el promedio de IPLR por cada servicio, la **Tabla 24** muestra la relación entre el promedio y la expresión de disponibilidad, donde se observa que todos los servicios de la red estuvieron disponibles durante la prueba realizada.

<sup>30</sup> La recomendación ITU-T Y.1540 (2019) establece que la disponibilidad de un servicio puede determinarse utilizando la expresión  $IPLR < c_1$ , donde  $c_1 = 0,20$  es el valor de referencia especificado por la ITU. Este criterio se aplicará en la evaluación de todos los escenarios propuestos en la sección 4.1.3, permitiendo así determinar la disponibilidad de los distintos servicios en función de la Tasa de Pérdida de Paquetes IP (IPRL).



**Tabla 24**  
*Disponibilidad de red en Escenario 1, sin QoS.*

Servicio	Expresión Disponibilidad $IPRL < c_1$	Disponibilidad
VoIP	$0,00425 < 0,20$	Si
VOD	$0,00431 < 0,20$	Si
FTP	$0,00427 < 0,20$	Si

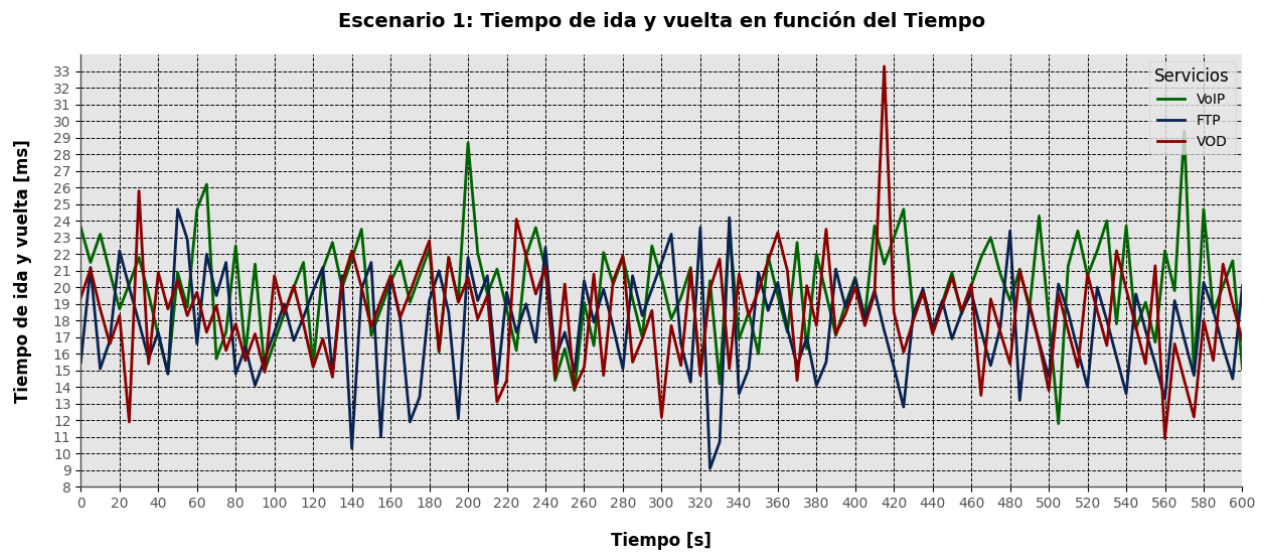
- **Tiempo de ida y vuelta (RTT)**

Los datos de RTT se recopilaban utilizando la herramienta Hping3. Para llevar a cabo la medición, se empleó el comando `sudo timeout 600 hping3 <dirección-ip-seridor> -p <puerto> -m 1500`. En este comando, el parámetro `timeout 600` define el tiempo de la prueba en 600 segundos. Cabe destacar que `timeout` no es un parámetro específico de Hping3, pues es una herramienta incluida en el paquete `coreutils` de los sistemas Linux, que permite limitar el tiempo de ejecución de un comando. Y el parámetro `-m 1500` define el tamaño del paquete en bytes.

En este contexto, durante la prueba con hping3, se enviaron paquetes a una dirección IP y puertos específicos, en donde se mide el RTT de cada paquete en milisegundos (ms). De esta forma se obtiene el RTT de 600 paquetes, tales datos se encuentran representados en la **Figura 83**, donde es posible visualizar que el RTT se encuentra por encima de los 9 ms y alcanza un valor máximo de 33.2 ms. Así, en este primer escenario, se evidencia que el RTT es casi estable, con pequeñas variaciones que pueden estar relacionadas con fluctuaciones normales del tráfico en una red sin priorización.

### Figura 83

Evaluación de Tiempo de ida y vuelta en escenario 1, sin QoS.



- **Retardo de transferencia de paquetes IP (IPTD)**

El retardo de transferencia de paquetes IP, también conocido como latencia unidireccional es el tiempo que un paquete tarda en viajar desde el origen hasta el destino. La medición directa de este KPI no es posible con las herramientas seleccionadas. Sin embargo, existen aproximaciones que se pueden realizar para conseguir un valor de referencia de este KPI. Según la recomendación ITU-T Y.1540 (2019), cuando no se disponen de mediciones directas del IPTD, una de las formas de aproximación es a partir del RTT. Dado que RTT permite conocer el tiempo total que un paquete tarda en ir del origen al destino y regresar, es posible aproximar el valor de la latencia unidireccional dividiendo el RTT entre dos, como se indica en la **Ec. 5**.

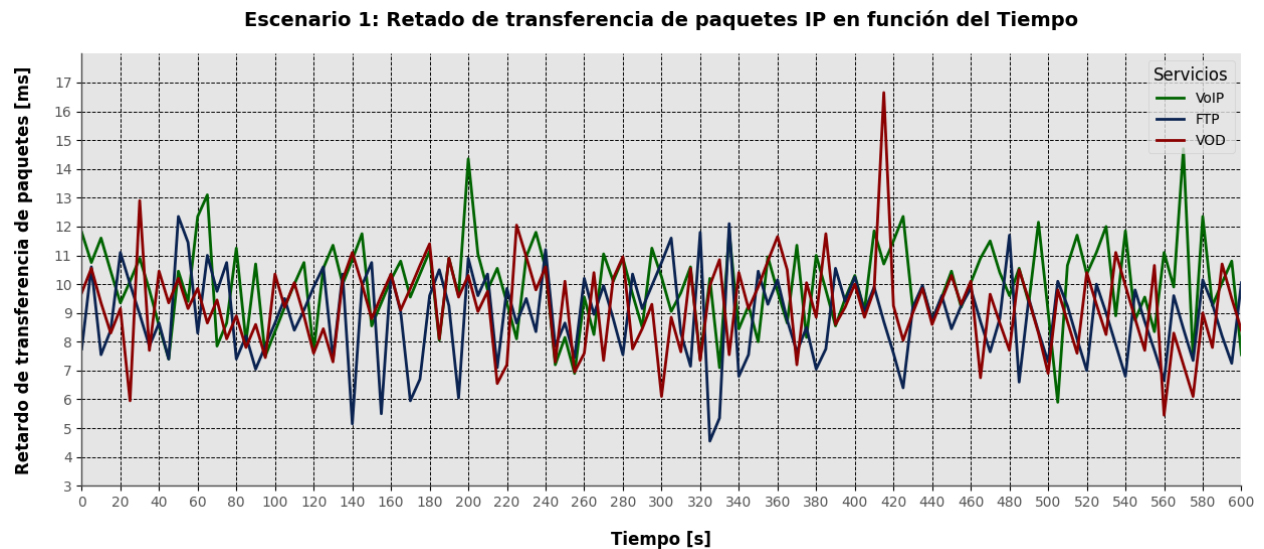
$$Latencia = \frac{RTT}{2} \quad (\text{Ec. 5}).$$

Si bien la aproximación de latencia no considera los factores que pueden presentarse en la transmisión de paquetes, proporciona una referencia útil para identificar tendencias y comportamientos en la red. De esta forma, se emplean los datos de RTT obtenidos en la prueba anterior y se realiza esta aproximación para obtener el valor de latencia. Con los datos aproximados se presenta la **Figura 84**, donde se observa que no hay valores menores a 4 ms y

el valor máximo que se alcanza es de 16,8 ms. Además, debido a que se empleó los datos de RTT para el cálculo de IPTD, la fluctuación en los valores de ambos KPIs es equivalente.

#### Figura 84

*Evaluación de IPTD en escenario 1, sin QoS.*



#### 4.1.3.1.2. Escenario 2: Tráfico moderado

La evaluación de este escenario presenta una población de interés que consiste en paquetes de 30000 bytes. De esta forma se evalúa y recopila datos en función de esta población de interés y los cinco KPIs establecidos. Finalmente, se presenta una gráfica que muestra la tendencia de los datos recopilados durante el periodo medido.

- **Ancho de Banda (AB)**

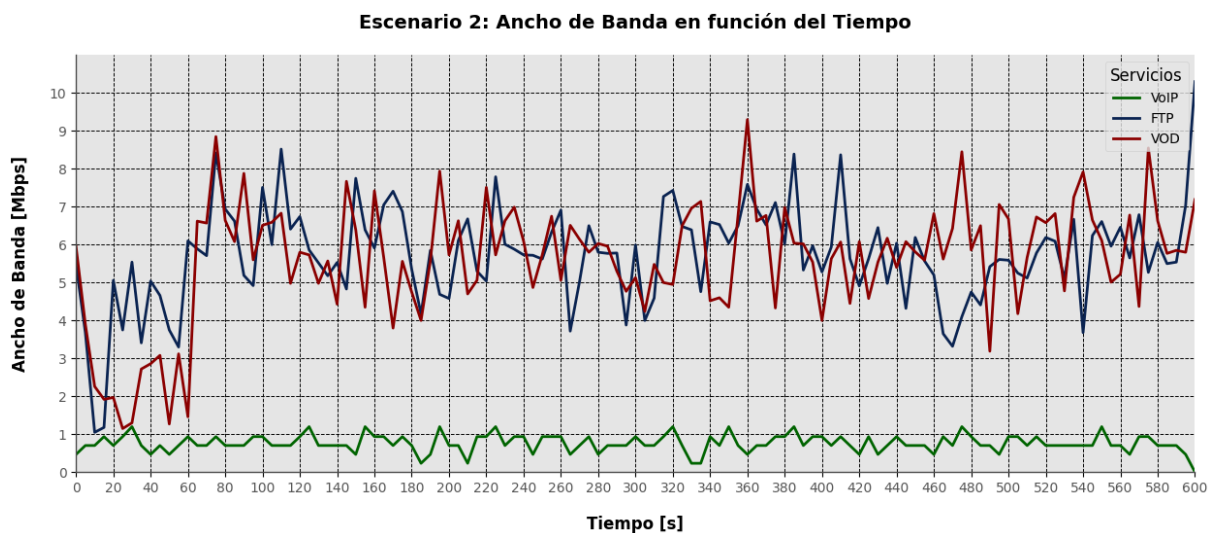
La medición de AB con iPerf3 requiere una configuración cliente-servidor. De esta forma en los servidores se emplea el comando `iperf3 -s - <puerto>` para que estén a la escucha de sus clientes y puedan presentar estadísticas de ello. Mientras que, en los clientes se utiliza el comando `iperf3 -c <direccion-ip-servidor> -p <puerto> -l 30000 -t 600`. Donde el parámetro `-l 30000` especifica el tamaño del paquete en bytes y `-t 600` indica la duración de la prueba en segundos. Cabe señalar que, para el caso del servidor que funciona bajo UDP, en el cliente se agrega el parámetro `-u` al comando anterior.

Con los datos recopilados, se obtiene la **Figura 85**, la cual muestra la fluctuación de AB en intervalos de cinco segundos. En esta figura, se puede observar que el valor máximo alcanzado es de 11 Mbps, correspondiente al servicio FTP. Por otro lado, el servicio VOD alcanza un valor máximo de 9.3 Mbps. Y de manera similar al escenario anterior, el servicio VoIP presenta valor bajos de AB, con un máximo de 1.2 Mbps.

El comportamiento de los datos en este escenario muestra una disminución considerable de ancho de banda respecto al escenario 1 (**Figura 78**). De acuerdo con Laassiri et al. (2017) en su trabajo “Evaluación de los parámetros QoS en diferentes arquitecturas SDN utilizando Omnet 4.6++”, esto sucede porque a medida que el tamaño de paquetes aumenta, es necesario fragmentarlos para poder transmitirlos, lo cual genera más tráfico en la red y por ende la disminución del AB.

**Figura 85**

*Evaluación de Ancho de banda en escenario 2, sin QoS.*



- **Tasa de paquetes perdidos (IPRL)**

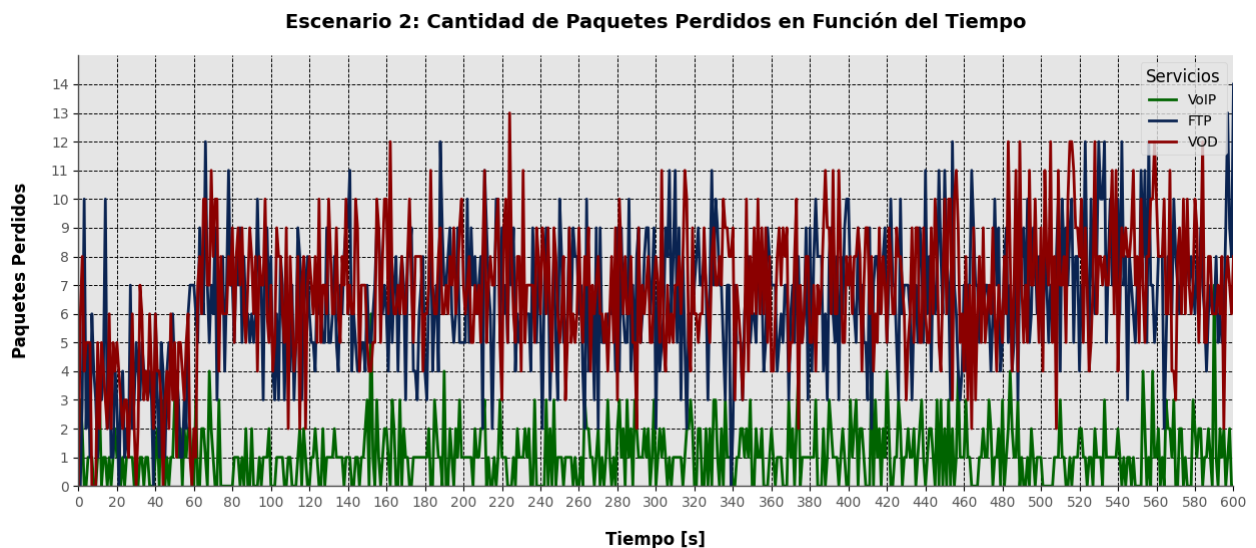
La cantidad de paquetes perdidos para el servicio VoIP se obtiene con el comando de iPerf3 presentado en la sección anterior. Y para el caso de los servicios que funcionan bajo el protocolo TCP, se emplea los siguientes filtros en Wireshark: `tcp.port==21 &&`

tcp.analysis.lost\_segment y tcp.port==8096 && tcp.analysis.lost\_segment, para los servicios FTP y VOD respectivamente.

La **Figura 86** muestra la cantidad de paquetes perdidos en función del tiempo, con un intervalo de representación es de un segundo, para visualizar todas las pérdidas de paquetes durante la prueba. En la figura se observa que todos los servicios presentan pérdidas significativas y a diferencia del escenario 1, en este escenario las pérdidas son más sucesivas, lo que indica mayor cantidad de pérdidas referente al aumento de tráfico.

### Figura 86

*Evaluación de Paquetes perdidos en escenario 2, sin QoS.*



Para identificar el servicio con la mayor cantidad de pérdidas, se obtiene el valor de IPRL, como la relación entre la cantidad de paquetes perdidos y la cantidad de paquetes transmitidos. Estos datos se obtienen a partir de las herramientas iPerf3 y Wireshark, como se muestra en las **Figuras 82, 83 y 84**. La **Tabla 25** presenta el IPRL de cada servicio, donde se observa que el servicio VoIP presenta la mayor cantidad de pérdidas, con un valor de IPRL de 0.24.

**Figura 87**  
Cantidad de paquetes perdidos y enviados para servicio VoIP.

```

[ 5] 596.00-597.00 sec 117 KBytes 960 Kbits/sec 4.123 ms 0/4 (0%)
[ 5] 597.00-598.00 sec 117 KBytes 960 Kbits/sec 3.977 ms 1/5 (20%)
[ 5] 598.00-599.00 sec 58.6 KBytes 480 Kbits/sec 3.584 ms 2/4 (50%)
[ 5] 599.00-600.00 sec 117 KBytes 960 Kbits/sec 3.391 ms 0/4 (0%)
[ 5] 600.00-600.06 sec 0.00 Bytes 0.00 bits/sec 3.391 ms 0/0 (0%)
-----
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total
Datagrams
[ 5] 0.00-600.06 sec 0.00 Bytes 0.00 bits/sec 3.391 ms 637/2621 (24%)
-----
Server listening on 16384
    
```

**Figura 88**  
Cantidad de paquetes perdidos y enviados para servicio VOD.

**Figura 89**  
Cantidad de paquetes perdidos y enviados para servicio FTP.

**Tabla 25**

*Datos de IPLR obtenidos en el Escenario 2, sin QoS.*

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	637	2621	0,2430
VOD	4023	269333	0,0149
FTP	3845	242213	0,0158

- **Disponibilidad de red**

La disponibilidad de red depende del valor de IPRL. Dónde si se cumple la expresión  $IPRL < c_1$  (siendo  $c_1 = 0,20$ ), se considera disponible. Conociendo el valor promedio de IPRL es de cada servicio, se obtiene la **Tabla 26**, dónde se relaciona dicho promedio con la expresión de disponibilidad. De esta manera, se observa que el servicio VoIP presentó instantes de indisponibilidad durante la prueba, pues el valor promedio de IPRL es mayor a  $c_1$ . Mientras que los demás servicios permanecieron disponibles.

**Tabla 26**

*Disponibilidad de red en Escenario 2, sin QoS.*

Servicio	Expresión Disponibilidad $IPRL < c_1$	Disponibilidad
VoIP	$0,2430 < 0,20$	No
VOD	$0,0149 < 0,20$	Si
FTP	$0,0158 < 0,20$	Si

- **Tiempo de ida y vuelta (RTT)**

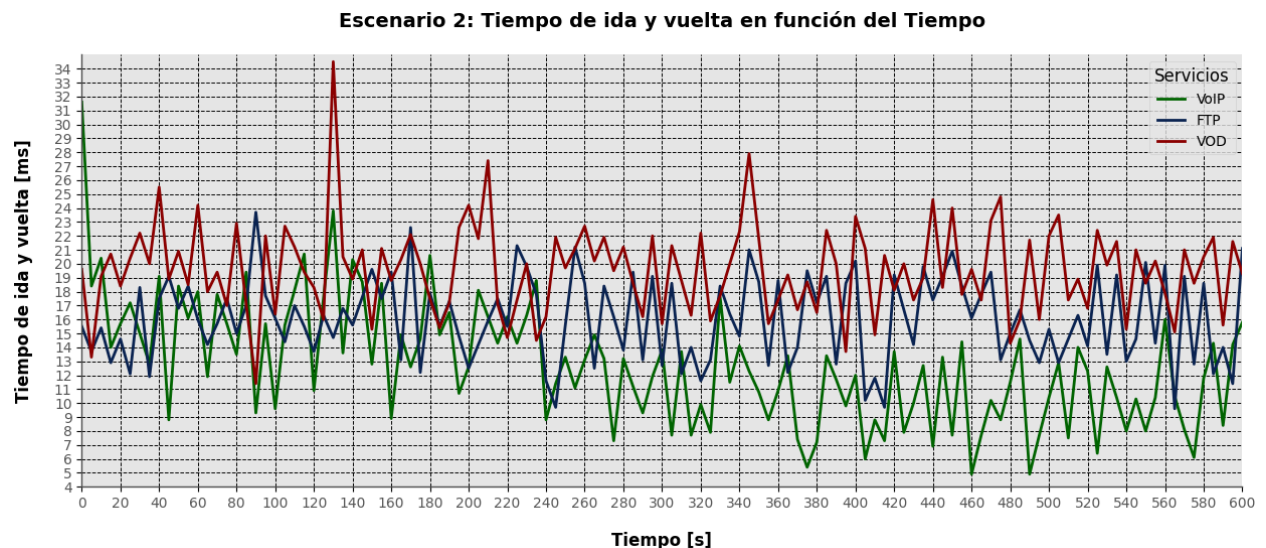
Para llevar a cabo la medición de RTT con Hping3, se empleó el comando `sudo timeout 600 hping3 <dirección-ip-servidor> -p <puerto> -m 30000`. En este comando, el parámetro `-m 30000` indica el tamaño de paquetes en bytes y con la herramienta `timeout` se determina la duración de 600 segundos para esta prueba. A partir de esta prueba, se obtuvieron los RTT de 600 paquetes, los cuales se representan en la **Figura 90**. En esta figura se observa



que el valor máximo de RTT es de 34,2 ms. En este caso, los datos de RTT presentan mayores fluctuaciones que en el escenario 1. Sin embargo, el valor máximo alcanzado indica un balance adecuado en la red, sin saturaciones críticas.

### Figura 90

*Evaluación de RTT en escenario 2, sin QoS.*



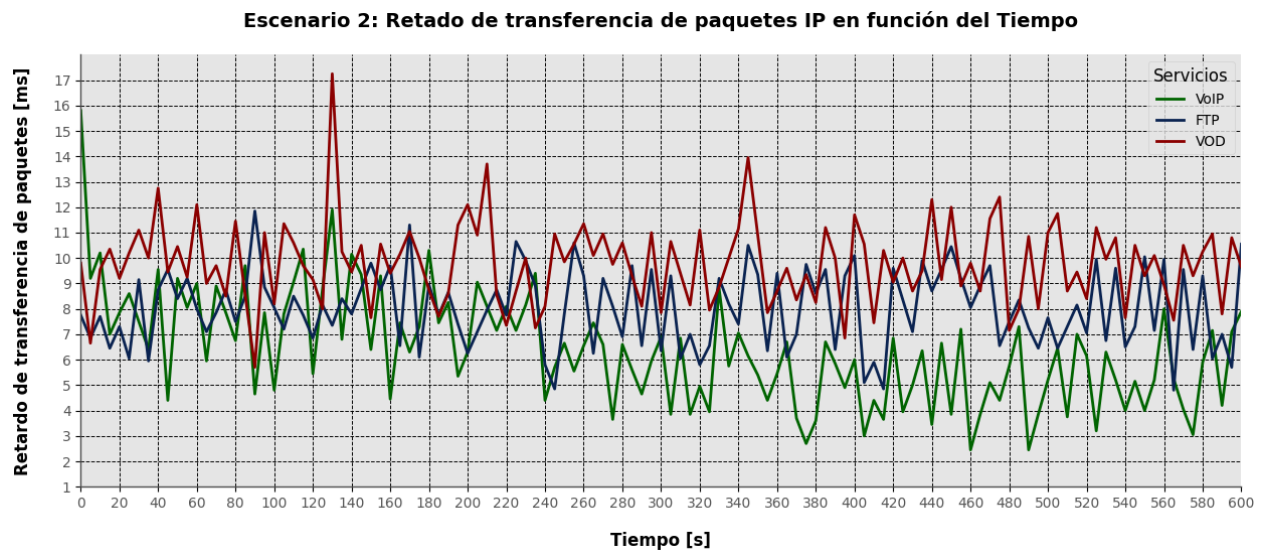
- **Retardo de transferencia de paquetes (IPTD)**

Empleando la **Ec. 5**, se obtuvo el valor de IPTD para los 600 paquetes analizados en la prueba. De esta forma, la **Figura 91** muestra el IPTD de los paquetes, donde es posible observar el valor máximo alcanzado es de 17,2 ms. La tendencia de los datos referentes a IPTD es proporcional a la de RTT, debido a la aproximación utilizada.



## Figura 91

Evaluación de IPTD en escenario 2, sin QoS.



### 4.1.3.1.3. Escenario 3: Tráfico intenso

En este escenario, se aumenta el MTU de los paquetes a 45000 bytes. Esto permite evaluar el impacto de la fragmentación en el rendimiento de la red. Se recopilan datos en función de los cinco KPI establecidos, y se presentan gráficas que muestran la fluctuación de estos parámetros a lo largo del tiempo.

- **Ancho de Banda (AB)**

En la configuración cliente-servidor con iPerf3 se emplean los siguientes comandos; para los servidores; `iperf3 -s <puerto>`, y para los clientes; `iperf3 -c <direccion-ip-servidor> -p <puerto> -l 45000 -t 600`, donde `-l 45000` establece el tamaño del paquete y con `-t 600` se establece la duración de la prueba en segundos. Además, para el caso del servidor que funciona bajo UDP, el cliente correspondiente debe agregar el parámetro `-u` al comando anterior.

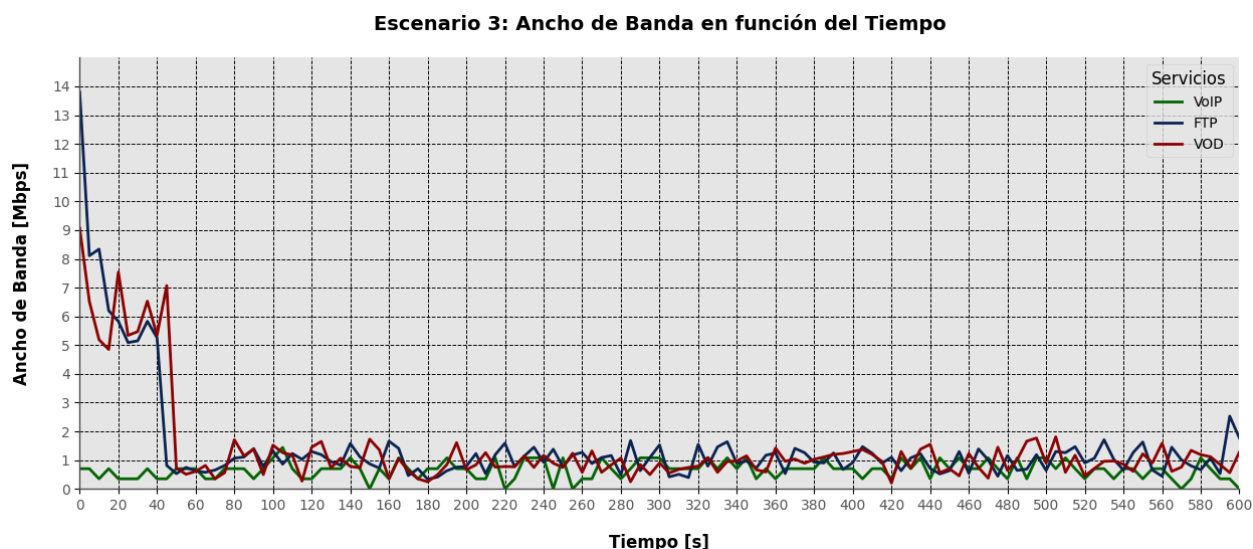
La **Figura 92** muestra los datos obtenidos de esta prueba. De esta forma se evidencia que, al principio de la prueba, los servicios VOD y FTP presentan picos entre 9 y 14 Mbps,

respectivamente, y desde el segundo 40 presentan un decremento significativo que oscila entre los 2 Mbps. Mientras que VoIP se mantiene en valores por debajo de 1 Mbps en toda la prueba.

Cranley & Davis (2005) indican que el comportamiento observado en los servicios VOD y FTP se debe a la sobrecarga de tráfico causada por la fragmentación de paquetes. Esta congestión genera altas tasas de retransmisión en los servicios que operan bajo el protocolo TCP, lo que provoca una disminución significativa en el uso del ancho de banda para estos servicios.

### Figura 92

*Evaluación de Ancho de banda en escenario 3, sin QoS.*



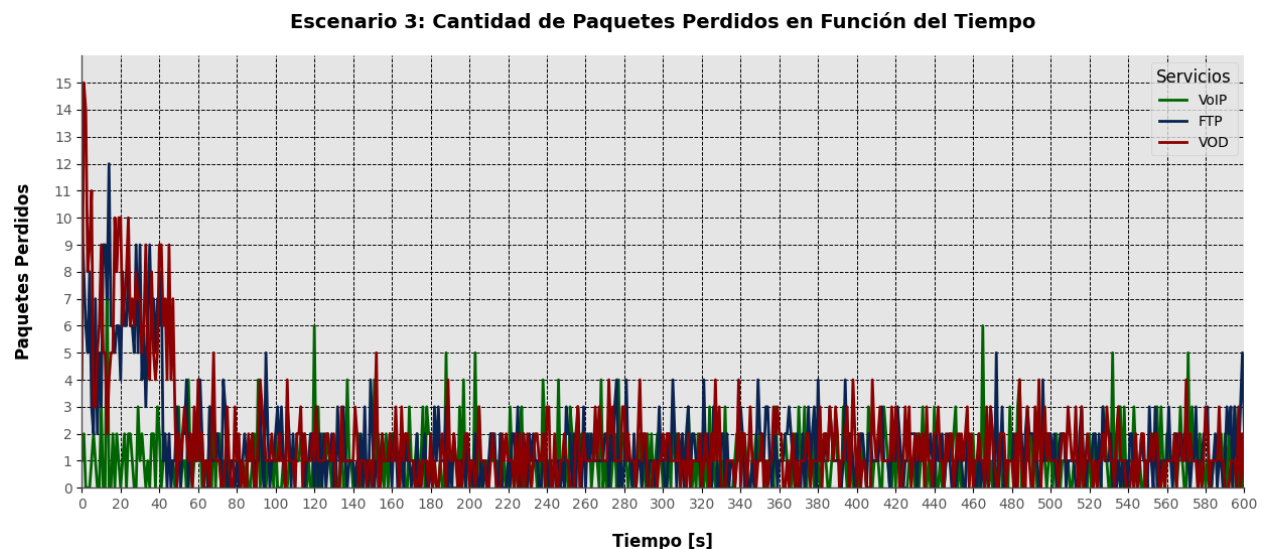
- **Tasa de paquetes perdidos (IPRL)**

Los datos sobre la cantidad de paquetes perdidos fueron recopilados con iPerf3 y los comandos expuestos en la sección anterior. Los datos sobre el servidor VoIP son presentados de forma clara con iPerf3, sin embargo, para los datos acerca de los servicios que funcionan bajo TCP, es necesario emplear los siguientes filtros en Wireshark: `tcp.port==21 && tcp.analysis.lost_segment` y `tcp.port==8096 && tcp.analysis.lost_segment`, para los servicios FTP y VOD respectivamente.

La **Figura 93** muestra la totalidad de paquetes perdidos durante la prueba de diez minutos, para cada uno de los servicios. Así, se observa que los servicios VOD y FTP presentan altos valores de paquetes perdidos en los primeros cuarenta segundos de la prueba. Mientras que las pérdidas de VoIP si bien no presentan valores altos, son más constantes en cada segundo de la prueba. De este modo, se determina que la cantidad de paquetes perdidos en este escenario es mayor que en escenarios anteriores, ya que se observan fluctuaciones continuas que indican pérdidas en múltiples instantes de tiempo.

### Figura 93

*Evaluación de Paquetes perdidos en escenario 3, sin QoS.*



Para conocer el valor de IPRL en el escenario evaluado, se recopiló los datos sobre la cantidad exacta de paquetes perdidos, utilizando las herramientas iPerf3 y Wireshark como se evidencia en las **Figuras 89, 90 y 91**. De este modo, la **Tabla 27** presenta el IPRL de cada servicio, donde se visualiza que VoIP presenta la mayor cantidad de pérdidas con un IPRL de 0,347.

**Figura 94**

*Cantidad de paquetes perdidos y enviados para servicio VoIP.*

```
[ 5] 596.00-597.00 sec 132 KBytes 1.08 Mbits/sec 61.261 ms 1/4 (25%)
[ 5] 597.00-598.00 sec 87.9 KBytes 720 Kbits/sec 79.313 ms 0/2 (0%)
[ 5] 598.00-599.00 sec 87.9 KBytes 721 Kbits/sec 69.890 ms 2/4 (50%)
[ 5] 599.00-600.00 sec 132 KBytes 1.08 Mbits/sec 60.924 ms 0/3 (0%)
[ 5] 600.00-600.06 sec 0.00 Bytes 0.00 bits/sec 60.924 ms 0/0 (0%)

-----
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5] 0.00-600.06 sec 0.00 Bytes   0.00 bits/sec 60.924 ms   607/1748 (35%)
-----
Server listening on 16384
-----
```

**Figura 95**

*Cantidad de paquetes perdidos y enviados para servicio VOD.*

**Figura 96**

*Cantidad de paquetes perdidos y enviados para servicio FTP.*

**Tabla 27**

Datos de IPLR obtenidos en el Escenario 3, sin QoS.

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	607	1748	0,3472
VOD	1010	95024	0,0106
FTP	951	93342	0,0101

- **Disponibilidad de red**

Conociendo el IPLR de cada servicio, se emplea el criterio  $IPLR < c_1$ , explicado en las anteriores secciones para determinar la disponibilidad del servicio durante la prueba realizada. Así, la **Tabla 28** muestra que el servicio VoIP presentó indisponibilidad en múltiples instantes de tiempo durante la prueba debido a que el valor de IPLR es mayor a  $c_1$ . Mientras que los servicios VOD y FTP si se encontraron disponibles.

**Tabla 28**

Disponibilidad de red en Escenario 3, sin QoS.

Servicio	Expresión Disponibilidad $IPLR < c_1$	Disponibilidad
VoIP	$0,3472 < 0,20$	No
VOD	$0,0106 < 0,20$	Si
FTP	$0,0102 < 0,20$	Si

- **Tiempo de ida y vuelta (RTT)**

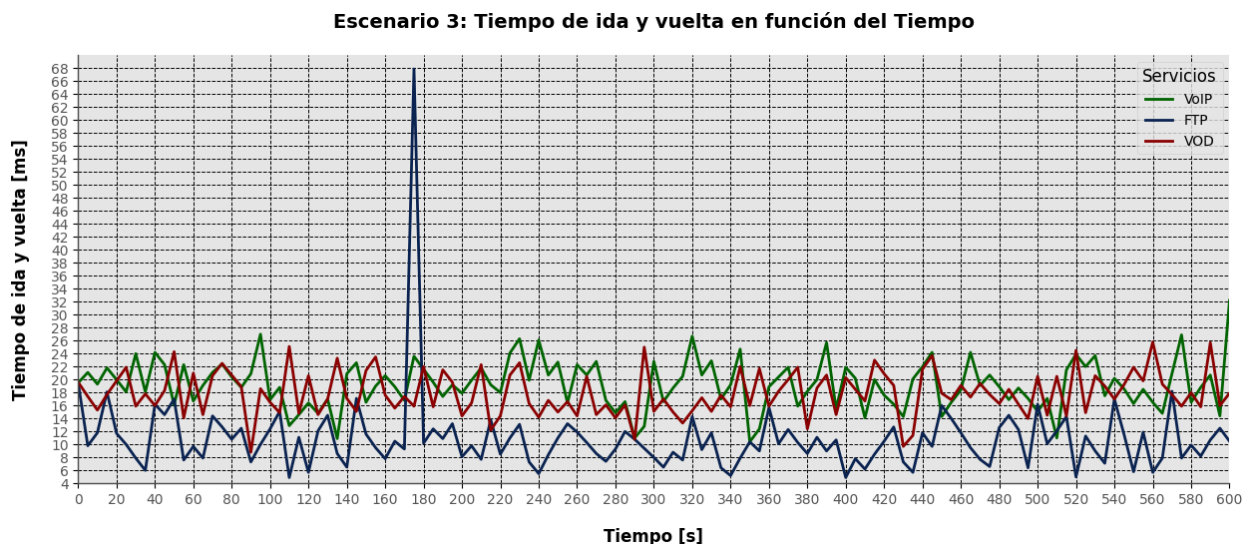
Para recopilar datos sobre RTT utilizando la herramienta Hping3 se emplea el comando `sudo timeout 600 hping3 <dirección-ip-servidor> -p <puerto> -m 45000`. En este comando, el parámetro `-m 45000` indica el tamaño de paquetes en bytes y con la herramienta `timeout` se define el tiempo de duración de la prueba en 600 segundos. Con los datos recopilados se obtiene la **Figura 97**, donde se observa que el RTT para el servicio FTP fluctúa entre los 4 y 20 ms, sin embargo, en el segundo 165 se observa un pico alto de 68 ms. El

servicio VOD fluctúa entre los valores de 10 a 26 ms. Y el servicio VoIP mantiene su RTT entre los valores de 12 a 32 ms.

Laassiri et al. (2017) afirman que, en redes WLAN sin QoS, cuando hay una sobrecarga de tráfico, los paquetes experimentan mayores tiempos de espera en las colas de almacenamiento temporal del punto de acceso antes de ser transmitidos, lo que incrementa el RTT. Así, la tendencia observada en el servicio FTP en el segundo 165 se debe a la alta densidad de tráfico en la red.

### Figura 97

*Evaluación de RTT en escenario 3, sin QoS.*

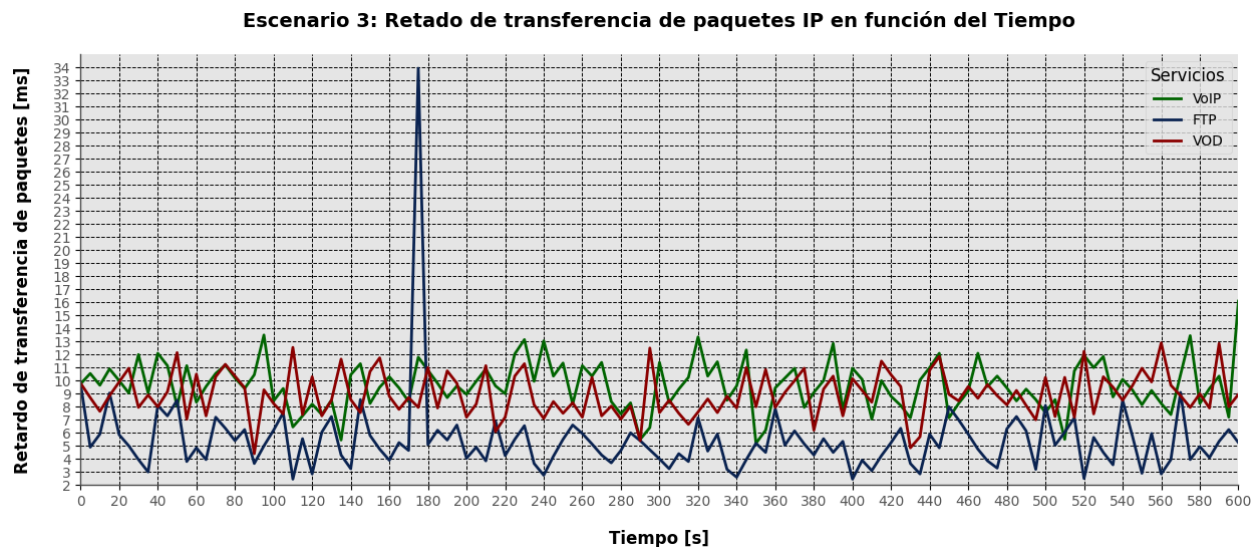


- **Retardo de transferencia de paquetes (IPTD)**

Al igual que en escenarios anteriores, se emplea la **Ec. 5** para obtener el valor de IPTD. De esta forma, luego de realizar los cálculos correspondientes y obtener los datos necesarios, se realiza la **Figura 98**, donde se observa que la fluctuación del IPTD de los servicios se encuentra entre los valores de 3ms a 16ms. Al igual que en escenarios anteriores, la tendencia de los datos de IPTD permanece igual a la de RTT.

**Figura 98**

*Evaluación de IPTD en escenario 3, sin QoS.*



#### 4.1.3.1.4. Escenario 4: Sobrecarga de red

En este escenario, el MTU de los paquetes es de 65507 bytes. Aunque el valor máximo permitido para el MTU es de 65535 bytes, las herramientas de evaluación utilizadas no ofrecieron soporte para dicha configuración. Por esta razón, se adoptó el límite máximo permitido por estas herramientas: 65,507 bytes. A partir de esta configuración, se evalúan y recopilan datos en función de los cinco KPIs establecidos y se presenta un gráfico de ellos.

- **Ancho de Banda (AB)**

Para medir el ancho de banda con iPerf3 se emplean los siguientes comandos; en los servidores se ejecuta el comando `iperf3 -s <puerto>`; mientras que en los clientes se ejecuta el comando `iperf3 -c <direccion-ip-servidor> -p <puerto> -l 65506 -t 600`. Dónde el parámetro `-l 65507` especifica el tamaño del paquete en bytes. Así se evalúan y recopilan durante los diez minutos de la prueba.

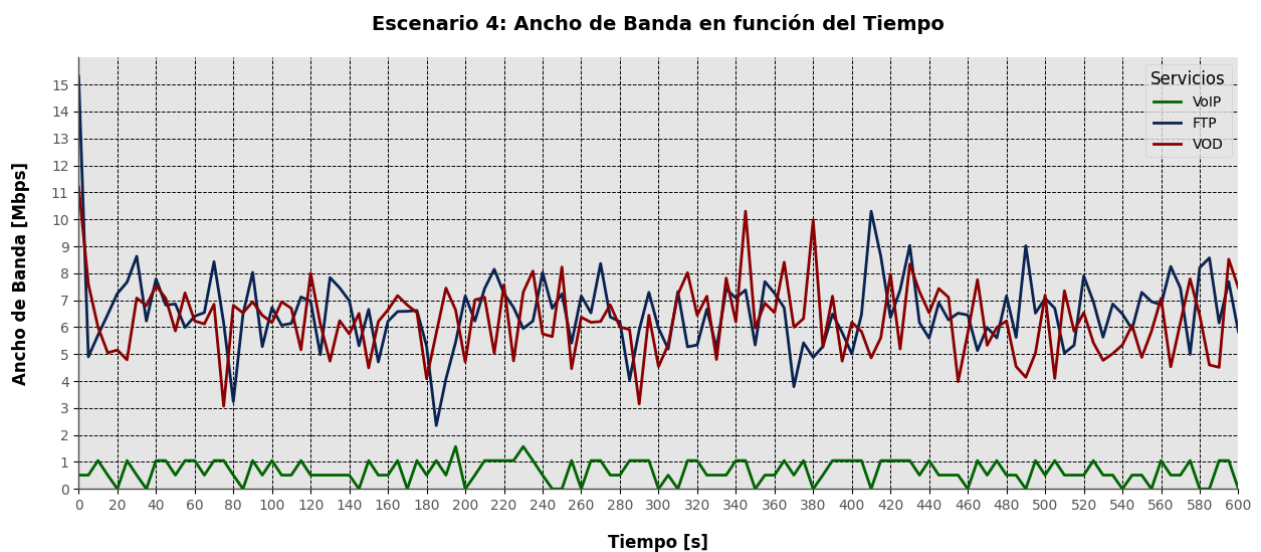
Los datos obtenidos se ilustran en la **Figura 94**, dónde se observa que el ancho de banda fluctúa entre 2 y 10 Mbps para el caso del servicio VoIP. El servicio VOD se encuentra entre los valores 3 y 10 Mbps. Y el servicio VoIP se mantiene en valores menos a 2 Mbps. Cabe



destacar que, en este último escenario, al trabajar con un MTU de 65507 bytes, la carga de tráfico es significativamente mayor en comparación con los escenarios anteriores. Como se muestra en la **Figura 99**, todos los servicios presentan fluctuaciones notables durante los diez minutos de prueba realizados. En particular, el servicio VoIP muestra frecuentes intervalos en los que el ancho de banda cae a 0 Mbps, lo que impacta de manera significativa en su funcionamiento del servicio.

**Figura 99**

*Evaluación de Ancho de banda en escenario 4, sin QoS.*



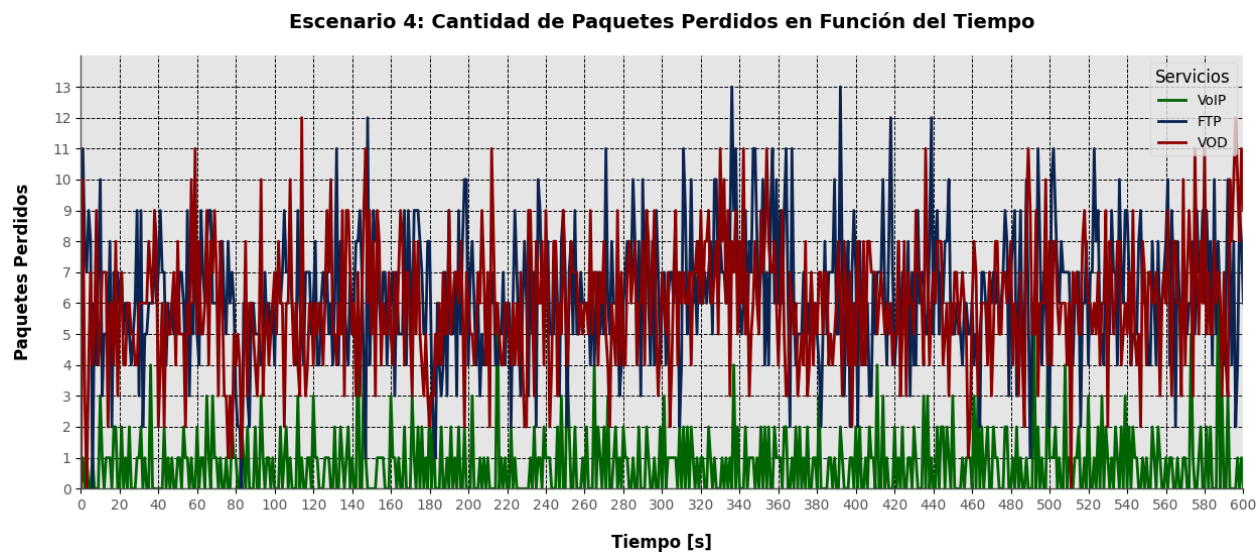
- **Tasa de paquetes perdidos (IPRL)**

La cantidad de paquetes perdidos se obtiene del escenario cliente-servidor con iPerf3, descrito en el apartado anterior. Para el caso de los servicios que funcionan con el protocolo TCP, se emplean los siguientes filtros de Wireshark: `tcp.port==21 && tcp.analysis.lost_segment` y `tcp.port==8096 && tcp.analysis.lost_segment`. Así, los datos obtenidos se ilustran en la **Figura 95**, dónde se evidencia la totalidad de paquetes perdidos por cada servicio durante cada segundo de la prueba. Asimismo, se observa que, durante el tiempo de prueba, las pérdidas fueron continuas y es por ello que la fluctuación de estas es casi imperceptible.



**Figura 100**

*Evaluación de Paquetes perdidos en escenario 4, sin QoS.*



De las herramientas iPerf3 y Wireshark se obtiene la cantidad de paquetes enviados y perdidos como se evidencia en las **Figuras 96, 97 y 98**, así se calcula el IPRL de cada servicio, como se muestra en la **Tabla 31**.

**Figura 101**

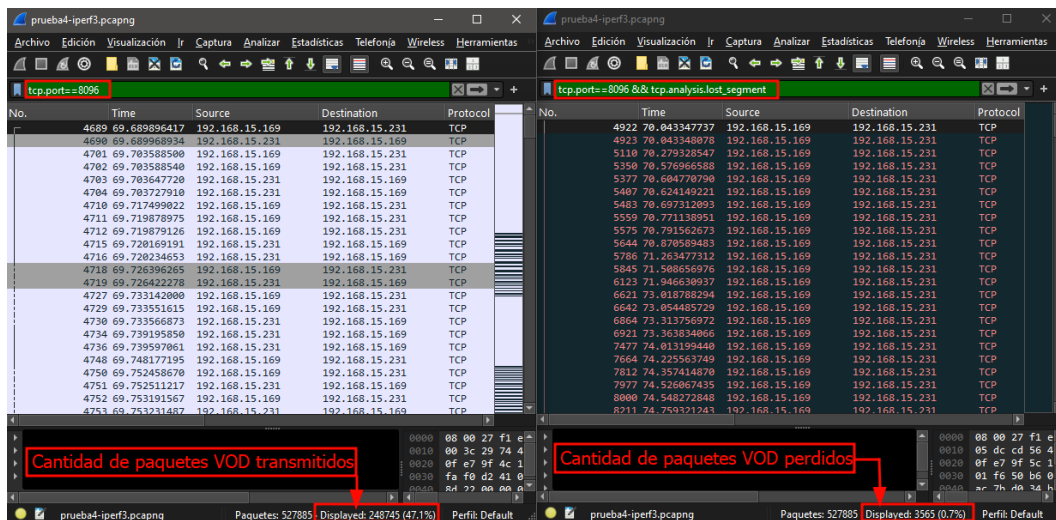
*Cantidad de paquetes perdidos y enviados para servicio VoIP.*

```

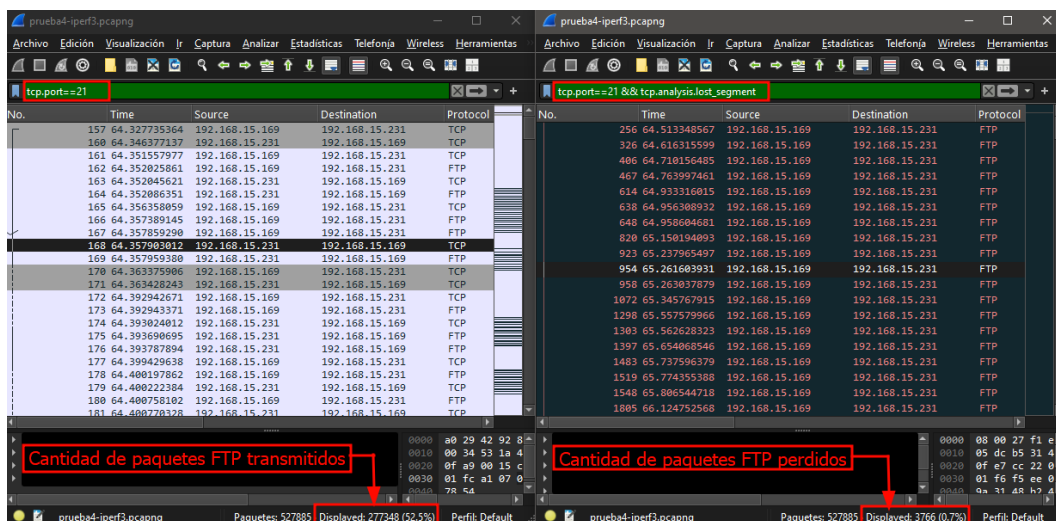
[ 5] 596.00-597.00 sec 128 KBytes 1.05 Mbits/sec 7.092 ms 0/2 (0%)
[ 5] 597.00-598.00 sec 64.0 KBytes 524 Kbits/sec 7.235 ms 1/2 (50%)
[ 5] 598.00-599.00 sec 128 KBytes 1.05 Mbits/sec 7.189 ms 0/2 (0%)
[ 5] 599.00-600.00 sec 64.0 KBytes 524 Kbits/sec 7.501 ms 1/2 (50%)
[ 5] 600.00-600.05 sec 0.00 Bytes 0.00 bits/sec 7.501 ms 0/0 (0%)
-----
[ ID] Interval          Transfer      Bandwidth      Jitter          Lost/Total Datagrams
[ 5] 0.00-600.05 sec 0.00 Bytes 0.00 bits/sec 7.501 ms 466/1201 (39%)
-----
Server listening on 16384

```

**Figura 102**  
Cantidad de paquetes perdidos y enviados para servicio VOD.



**Figura 103**  
Cantidad de paquetes perdidos y enviados para servicio FTP.



**Tabla 29**  
Datos de IPLR obtenidos en el Escenario 4, sin QoS.

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	466	1201	0,3880
VOD	3565	248745	0,0143
FTP	3766	277348	0,0135

- **Disponibilidad de red**

En función de la expresión de disponibilidad, se emplea el IPRL de cada servicio para conocer si los mismos se encontraron o no disponibles en los diez minutos de prueba realizados. De esta forma, la **Tabla 32** muestra la relación de cada IPRL con la expresión de disponibilidad, dónde se observa que el servicio VoIP presentó indisponibilidad al igual que en los dos escenarios anterior. Mientras que los demás servicios si permanecieron disponibles.

**Tabla 30**

*Disponibilidad de red en Escenario 4, sin QoS.*

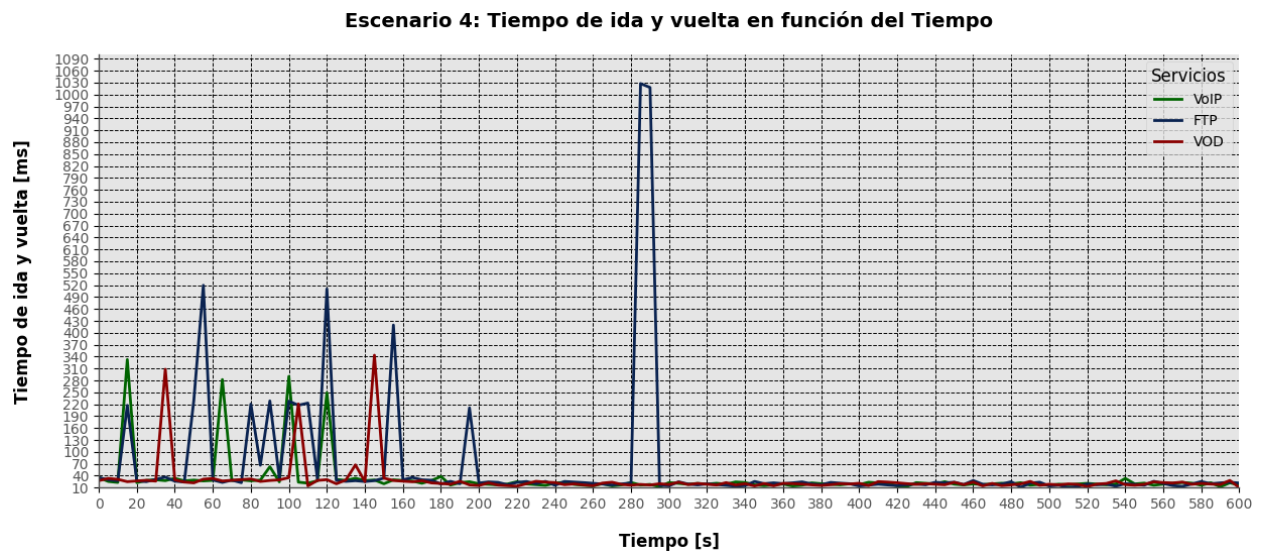
Servicio	Expresión Disponibilidad $IPRL < c_1$	Disponibilidad
VoIP	$0,3880 < 0,20$	No
VOD	$0,0143 < 0,20$	Si
FTP	$0,0135 < 0,20$	Si

- **Tiempo de ida y vuelta (RTT)**

Para medir el RTT se emplea la herramienta hping3 donde se ejecuta el comando `sudo timeout 600 hping3 <dirección-ip-servidor> -p <puerto> -m 65507`, dónde el parámetro `-m 65507` representa el MTU en bytes. Así, los datos obtenidos se encuentran representados en la **Figura 99**, dónde se evidencia que el RTT de los servicios fluctúa entre 10 y 520 ms. Además, se observa un poco alto que llega a 1030 ms, referente al servicio FTP. De esta manera, debido a la sobrecarga de tráfico generada por la fragmentación de paquetes, se observa un retardo significativo en el segundo 290. Tal como se explicó en el escenario 3, este retraso es el resultado del encolamiento sin prioridad dentro del AP.

**Figura 104**

*Evaluación de RTT en escenario 4, sin QoS.*

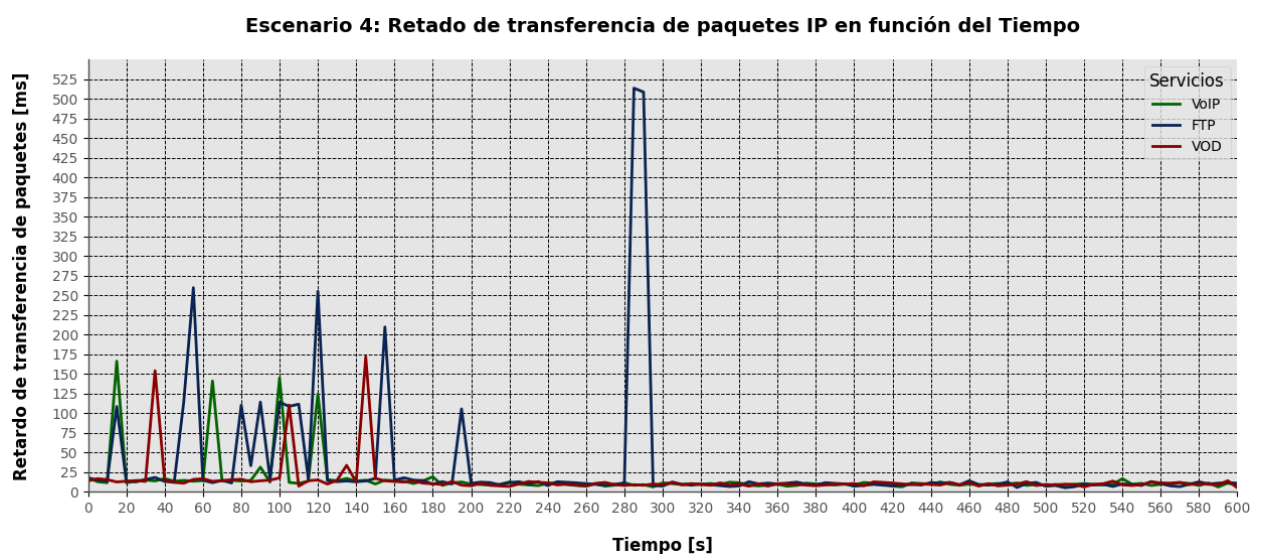


- **Retardo de transferencia de paquetes (IPTD)**

Considerando los valores de RTT, se calcula el IPTD y con dichos datos se realiza la **Figura 101**, dónde se visualiza la tendencia de IPTD de todos los servicios, destacando que hasta al inicio de la prueba los valores de RTT fluctúan entre 25 y 250 ms, mientras que después del segundo 200, los valores son inferiores a 25 ms. Además, la tendencia de los datos se mantiene al igual que RTT.

**Figura 105**

*Evaluación de IPTD en escenario 4, sin QoS.*



#### 4.1.3.2. Escenarios evaluados después de aplicar QoS

Las pruebas iniciales realizadas sin configuraciones de QoS en la red SDWN reflejan las limitaciones de una red sin mecanismos de priorización, donde los distintos flujos compiten por los recursos disponibles, lo que genera fluctuaciones y pérdidas significativas. En esta sección, se presenta la evaluación de los escenarios planteados, ahora con configuraciones de QoS en la red (ver **sección 3.3.2**). Estas políticas permiten una gestión más eficiente del tráfico, priorizando ciertos tipos de datos y mejorando el rendimiento de la red. Para esta evaluación, se emplearán los mismos comandos utilizados en la evaluación de escenarios sin QoS, por lo que no se profundizará en ellos. Además, tanto los datos obtenidos como los scripts utilizados para las gráficas están disponibles en el repositorio de GitHub: <https://github.com/knmoncayo/pruebas-sdwlan-tesis-utn.git>

##### 4.1.3.2.1. Escenario 1: Tráfico ligero

En este escenario se utiliza un tamaño de paquetes de 1500 bytes (MTU estándar). La evaluación emplea tres servidores iPerf3 y tres clientes. Cada dupla cliente-servidor opera en los puertos 16384/UDP, 8096/TCP y 21/TCP, que corresponden a los servicios de VoIP, VOD y FTP, respectivamente. Además, con la herramienta Hping3 también se analizará el tráfico de los puertos mencionados. Durante la evaluación, se recopilarán datos para graficarlos y analizar la tendencia de los cinco KPIs en función de las políticas de QoS configuradas.

- **Ancho de Banda (AB)**

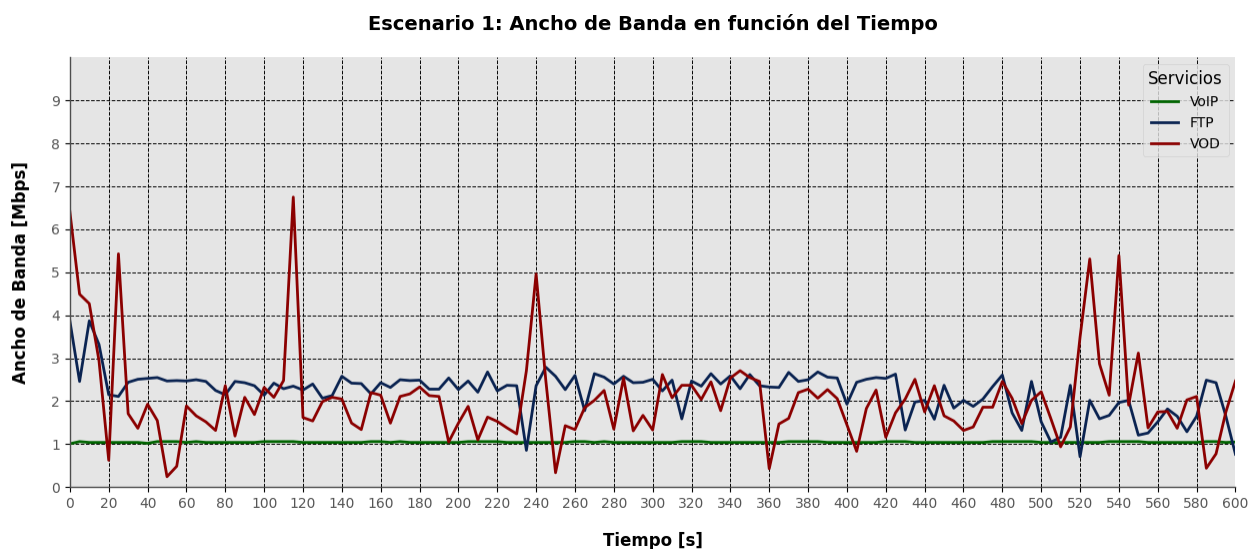
La **Figura 106** muestra los datos obtenidos en la evaluación del ancho de banda en la red con iPerf3. Con la implementación de QoS, se observa una asignación más eficiente del ancho de banda. Es posible apreciar que el servicio VoIP mantiene un ancho de banda constante de aproximadamente 1 Mbps, mientras que los servicios VOD y FTP presentan una utilización controlada sin afectar el rendimiento del servicio VoIP. Esto demuestra que cada servicio opera

dentro de los límites de ancho de banda máximo y mínimos configurados, evitando la saturación de la red con valores altos de ancho de banda (ver **Tabla 17**).

Las fluctuaciones observadas en los servicios VOD y FTP, a pesar de que el MTU de 1500 bytes evita la fragmentación en este escenario, se deben a factores relacionados con el comportamiento propio de estos servicios. Sin embargo, estas fluctuaciones no provocan una sobrecarga del ancho de banda en la red, a diferencia de lo registrado en la evaluación del primer escenario sin QoS (ver **Figura 78**).

### Figura 106

*Evaluación de Ancho de banda en escenario 1, con QoS.*

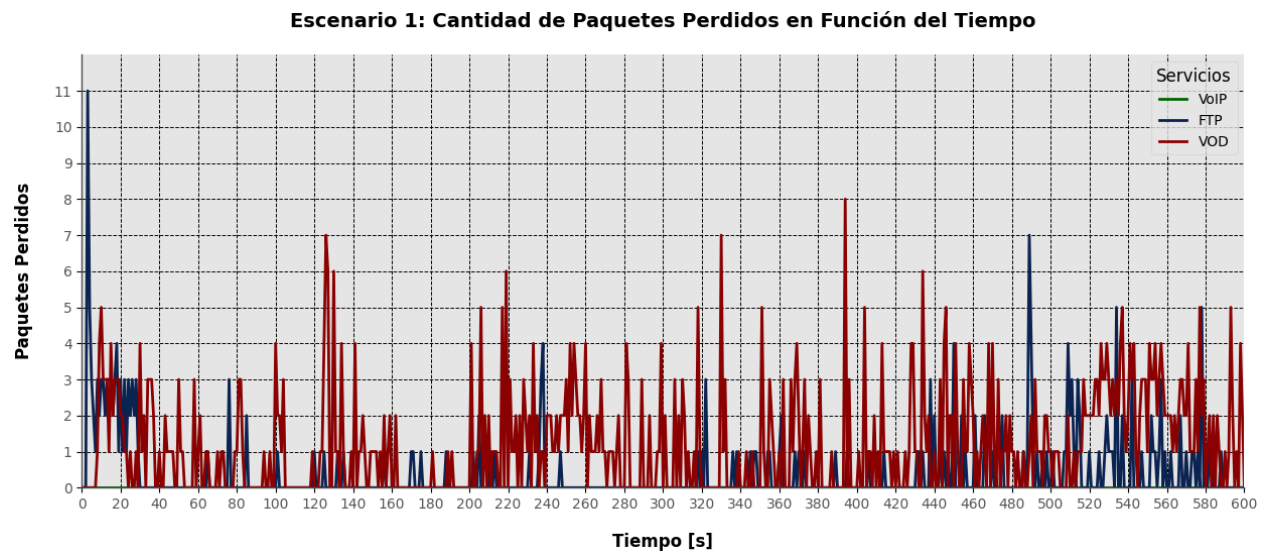


- **Tasa de paquetes perdidos (IPRL)**

La cantidad de paquetes perdidos demuestra el adecuado funcionamiento de las políticas de QoS configuradas. La **Figura 107** muestra que el servicio VoIP presenta un 0% de pérdidas, lo que garantiza una comunicación estable y sin interrupciones. Esto se debe a que VoIP es el servicio con mayor prioridad en la red. Por otro lado, aunque los servicios FTP y VOD experimentan pérdidas, estas son significativamente menores en comparación con las obtenidas en el escenario sin QoS (ver **Figura 79**).

**Figura 107**

*Evaluación de paquetes perdidos en escenario 1, con QoS.*



Las herramientas iPerf3 y Wireshark permiten analizar la cantidad de paquetes enviados y perdidos, como se muestra en las **Figuras 108, 109 y 110**. A partir de estos datos, se calcula la Tasa de Pérdidas de Paquetes IP (IPLR) para cada servicio (ver **Tabla 31**), donde se observa que los servicios VOD y FTP presentan valores muy cercanos, mientras que el servicio VoIP mantiene una tasa de pérdidas de cero.

**Figura 108**

*Cantidad de paquetes perdidos y enviados para servicio VoIP.*

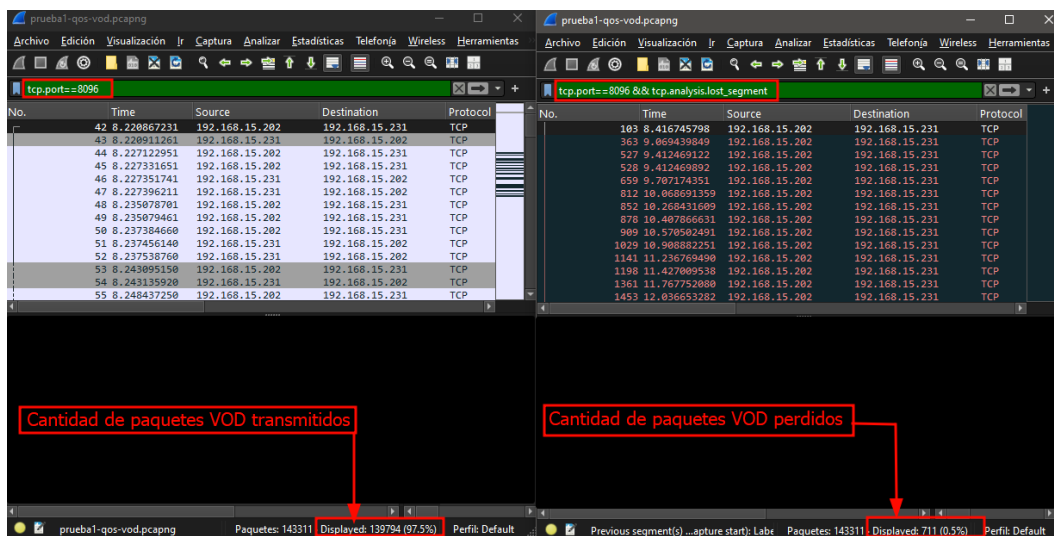
```

[ 5] 596.00-597.00 sec 129 KBytes 1.06 Mbits/sec 0.663 ms 0/88 (0%)
[ 5] 597.00-598.00 sec 127 KBytes 1.04 Mbits/sec 1.038 ms 0/87 (0%)
[ 5] 598.00-599.00 sec 129 KBytes 1.06 Mbits/sec 0.677 ms 0/88 (0%)
[ 5] 599.00-600.00 sec 127 KBytes 1.04 Mbits/sec 1.962 ms 0/87 (0%)
[ 5] 600.00-600.05 sec 5.86 KBytes 1.05 Mbits/sec 1.815 ms 0/4 (0%)
-----
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5]  0.00-600.05 sec 0.00 Bytes   0.00 bits/sec  1.815 ms   0/52429 (0%)
-----
Server listening on 16384
-----

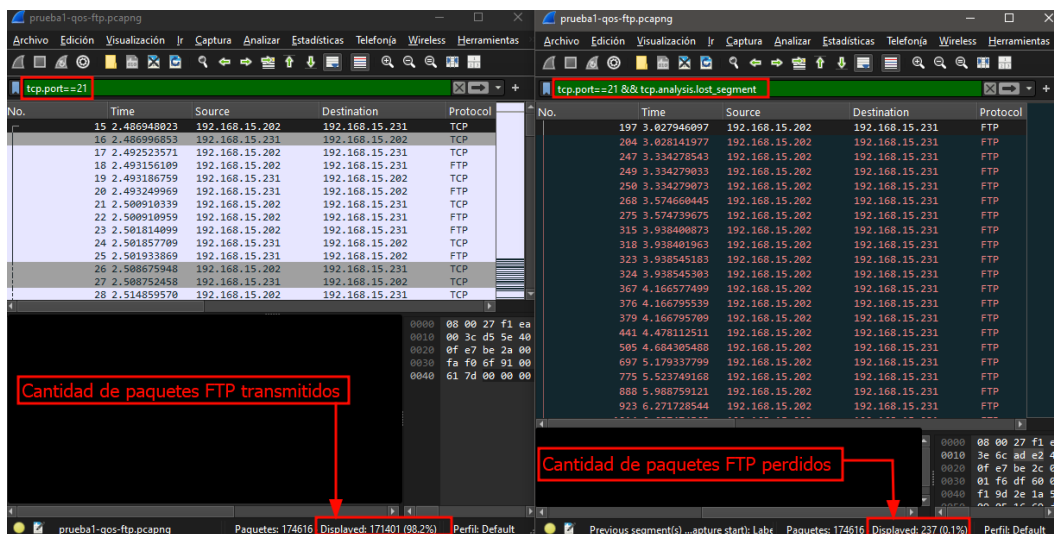
```



**Figura 109**  
Cantidad de paquetes perdidos y enviados para servicio VOD.



**Figura 110**  
Cantidad de paquetes perdidos y enviados para servicio FTP.



**Tabla 31**  
Datos de IPLR obtenidos en el Escenario 1, con QoS.

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	0	52429	0
VOD	711	139794	0,0050
FTP	237	171401	0,0013



- **Disponibilidad de red**

La **Tabla 32** muestra que todos los servicios cumplen con el criterio de disponibilidad definido en la Recomendación ITU-T Y.1540, donde el valor de IPRL debe cumplir con la expresión:  $IPLR < 0,20$ . Esto indica que la red se mantuvo disponible para todos los servicios durante la prueba realizada.

**Tabla 32**

*Disponibilidad de red en Escenario 1, con QoS.*

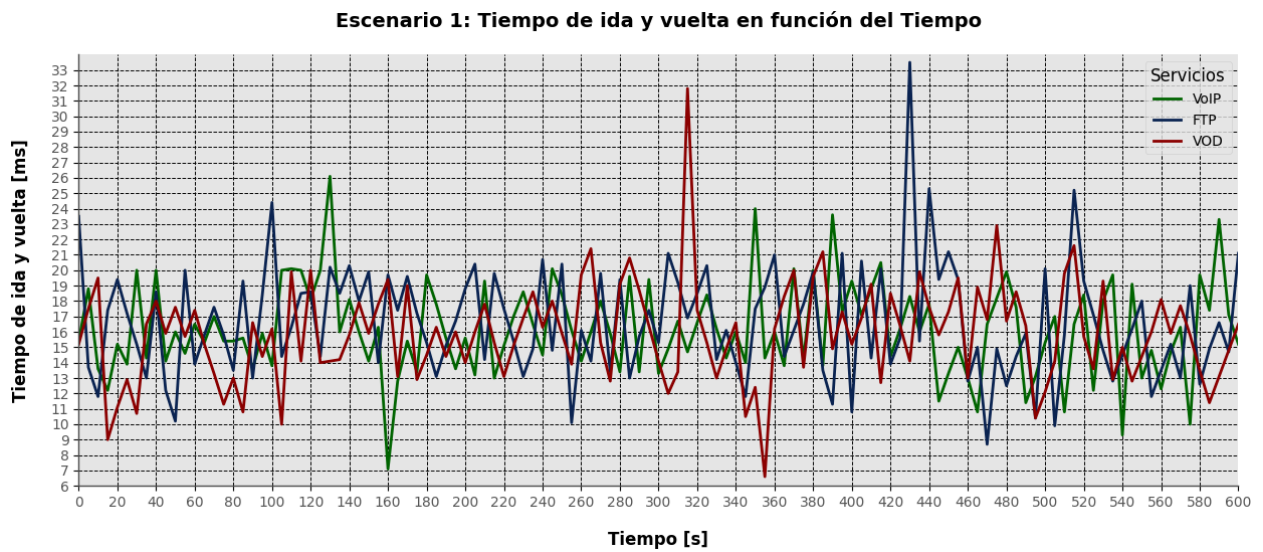
Servicio	Expresión Disponibilidad $IPLR < c_1$	Disponibilidad
VoIP	$0 < 0,20$	Si
VOD	$0,0050 < 0,20$	Si
FTP	$0,0013 < 0,20$	Si

- **Tiempo de ida y vuelta (RTT)**

En la evaluación de RTT se emplea la herramienta Hping3. La **Figura 111** muestra que el servicio VoIP presenta valores estables y bajo los 20 ms, lo cual es fundamental para aplicaciones en tiempo real. Asimismo, los servicios VOD y FTP muestran una notable estabilidad durante la mayor parte de la prueba.

**Figura 111**

Evaluación de Tiempo de ida y vuelta en escenario 1, con QoS.



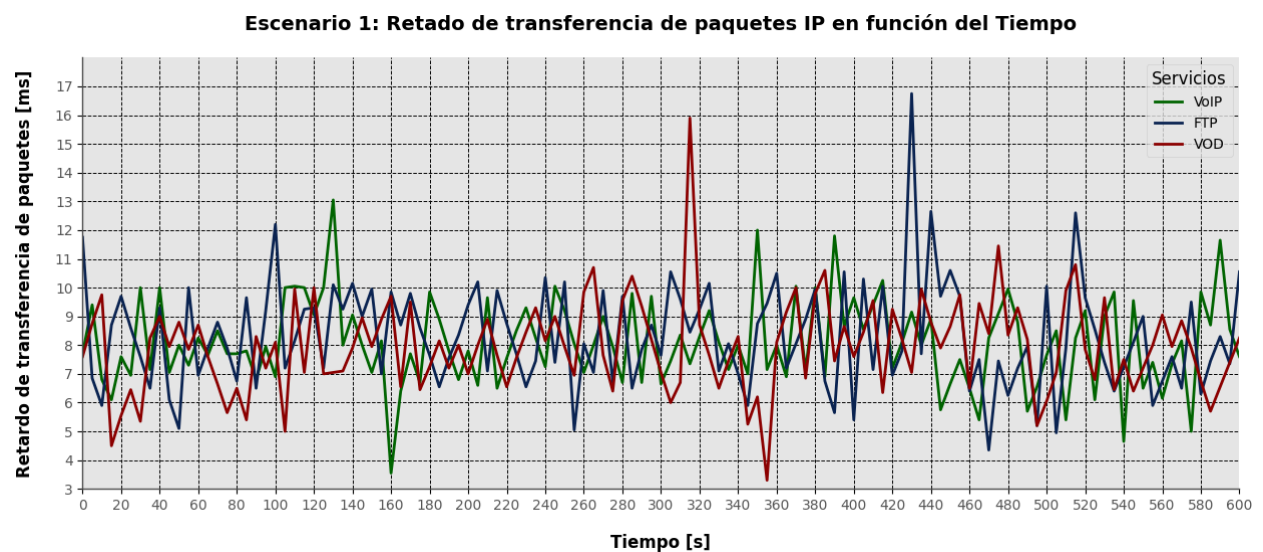
- **Retardo de transferencia de paquetes (IPTD)**

En este y los escenarios siguientes, también se emplea la aproximación  $latencia = \frac{RTT}{2}$ .

De este modo, los datos obtenidos se presentan en la **Figura 112**, donde se observa que todos los servicios mantienen valores inferiores a 10 ms, asegurando una respuesta adecuada para aplicaciones en tiempo real como VoIP.

**Figura 112**

Evaluación de IPTD en escenario 1, con QoS.



#### 4.1.3.2.2. Escenario 2: Tráfico moderado

En este escenario, el tamaño de paquetes se incrementó a 30000 bytes con el fin de evaluar su impacto en el rendimiento de la red con políticas de QoS. Durante la evaluación, se recopilaron y graficaron los datos correspondientes a los cinco KPIs definidos para observar cómo se comporta la red bajo estas condiciones específicas.

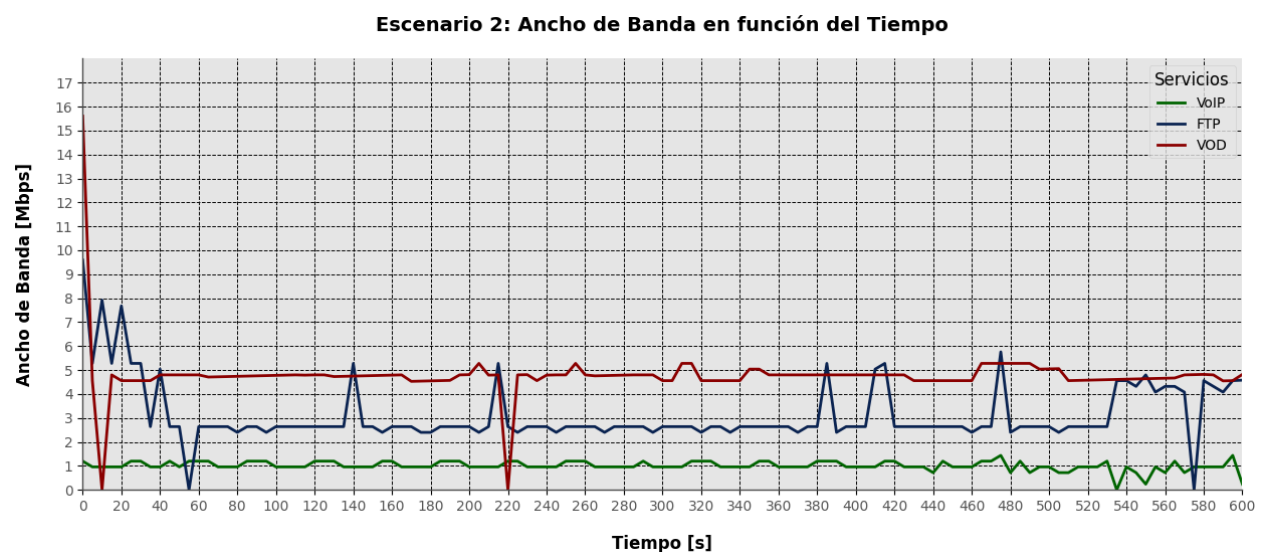
- **Ancho de Banda (AB)**

La **Figura 113** presenta los resultados de la evaluación de ancho de banda utilizando iPerf3. En esta gráfica se evidencia que la implementación de QoS garantizó una asignación eficiente de los recursos de la red para todos servicios. Cabe destacar que, debido al alto valor del MTU configurado, los paquetes se fragmentan durante la transmisión, proceso que se repetirá en los siguientes escenarios.

La gráfica muestra que el ancho de banda de VoIP se mantiene constante debido a la alta prioridad otorgada en la configuración de QoS. Mientras que los servicios VOD y FTP presentan algunos picos, donde FTP es el más afectado debido a que es el servicio con la prioridad más prioridad en la red, lo que resulta en mayores fluctuaciones para este servicio.

### Figura 113

*Evaluación de Ancho de banda en escenario 2, con QoS.*

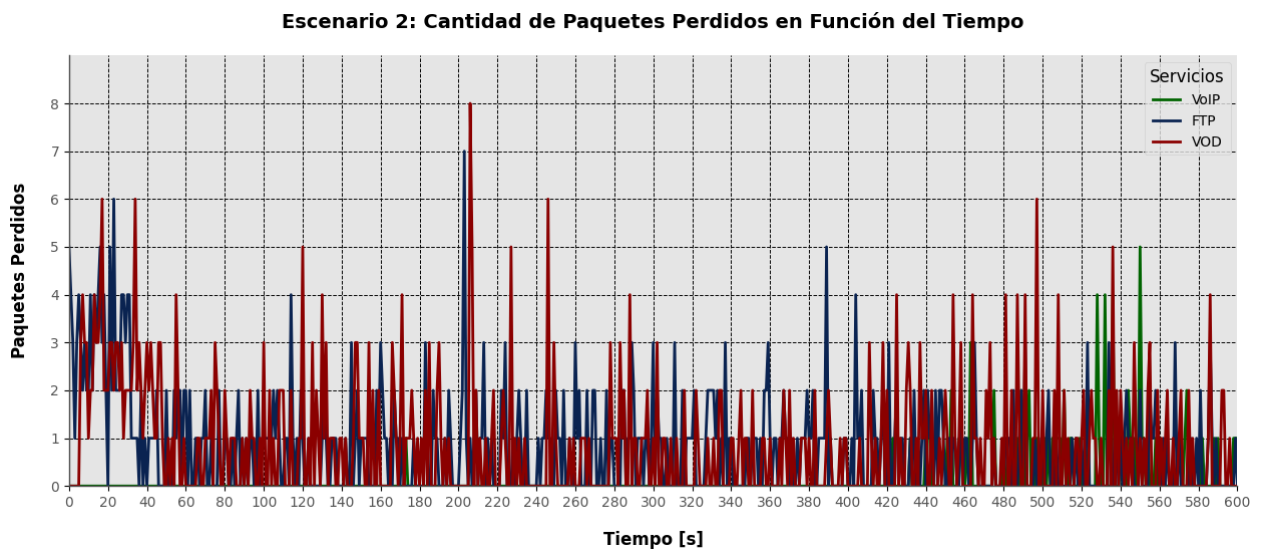


- Tasa de paquetes perdidos (IPRL)

La **Figura 114** muestra los resultados de la evaluación de pérdida de paquetes. Se observa que el servicio VoIP comienza a experimentar pérdidas a partir del segundo 173; sin embargo, estas pérdidas son mínimas y no afectan la calidad de las llamadas. Por otro lado, los servicios VOD y FTP presentan pérdidas durante gran parte de la prueba, donde VOD registra una mayor cantidad de paquetes perdidos en comparación con FTP.

### Figura 114

*Evaluación de Paquetes perdidos en escenario 2, con QoS.*



La evaluación con iPerf3 permite conocer la cantidad de paquetes enviados y perdidos. Para los servicios que operan con UDP, estos datos se registran en el terminal donde se ejecuta iPerf3. En cambio, para los servicios basados en TCP, se emplea Wireshark. Así, se obtienen las **Figuras 115, 116 y 117**. Con estos datos se calcula la tasa de paquetes perdidos como se muestra en la **Tabla 33**, donde se evidencia que el servicio VoIP presenta la tasa de paquetes perdidos más alta.

**Figura 115**  
Cantidad de paquetes perdidos y enviados para servicio VoIP.

```

[ 5] 596.00-597.00 sec 117 KBytes 961 Kbits/sec 66.731 ms 0/4 (0%)
[ 5] 597.00-598.00 sec 87.9 KBytes 720 Kbits/sec 80.582 ms 0/3 (0%)
[ 5] 598.00-599.00 sec 58.6 KBytes 480 Kbits/sec 107.597 ms 1/3 (33%)
[ 5] 599.00-600.00 sec 87.9 KBytes 720 Kbits/sec 112.892 ms 1/4 (25%)
[ 5] 600.00-601.00 sec 29.3 KBytes 240 Kbits/sec 137.496 ms 1/2 (50%)
[ 5] 601.00-601.23 sec 58.6 KBytes 2.07 Mbits/sec 138.915 ms 0/2 (0%)
-----
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5] 0.00-601.23 sec 0.00 Bytes   0.00 bits/sec  138.915 ms  77/2622 (2.9%)
-----
Server listening on 16384
    
```

**Figura 116**  
Cantidad de paquetes perdidos y enviados para servicio VOD.

Cantidad de paquetes VOD transmitidos

Cantidad de paquetes VOD perdidos

**Figura 117**  
Cantidad de paquetes perdidos y enviados para servicio FTP.

Cantidad de paquetes FTP transmitidos

Cantidad de paquetes FTP perdidos

**Tabla 33**  
*Datos de IPLR obtenidos en el Escenario 2, con QoS.*

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	77	2622	0,0293
VOD	522	151188	0,0034
FTP	503	220997	0,0023

- **Disponibilidad de red**

En la **Tabla 34** muestra que los valores de IPLR para todos los servicios evaluados son inferiores al valor de referencia 0,20. De esta manera, los servicios cumplen con los requisitos de disponibilidad establecidos por la ITU-T Y.1540. Esto indica que la red mantiene un rendimiento adecuado y una baja tasa de pérdida de paquetes, garantiza el óptimo funcionamiento de los servicios.

Cabe destacar que, en la evaluación sin QoS, el servicio VoIP presentó indisponibilidad durante la prueba. Esto evidencia la importancia de QoS para asegurar un desempeño estable y eficiente en entornos con distintos niveles de tráfico.

**Tabla 34**  
*Disponibilidad de red en Escenario 2, con QoS.*

Servicio	Expresión Disponibilidad $IPLR < c_1$	Disponibilidad
VoIP	$0,0293 < 0,20$	Si
VOD	$0,0034 < 0,20$	Si
FTP	$0,0023 < 0,20$	Si

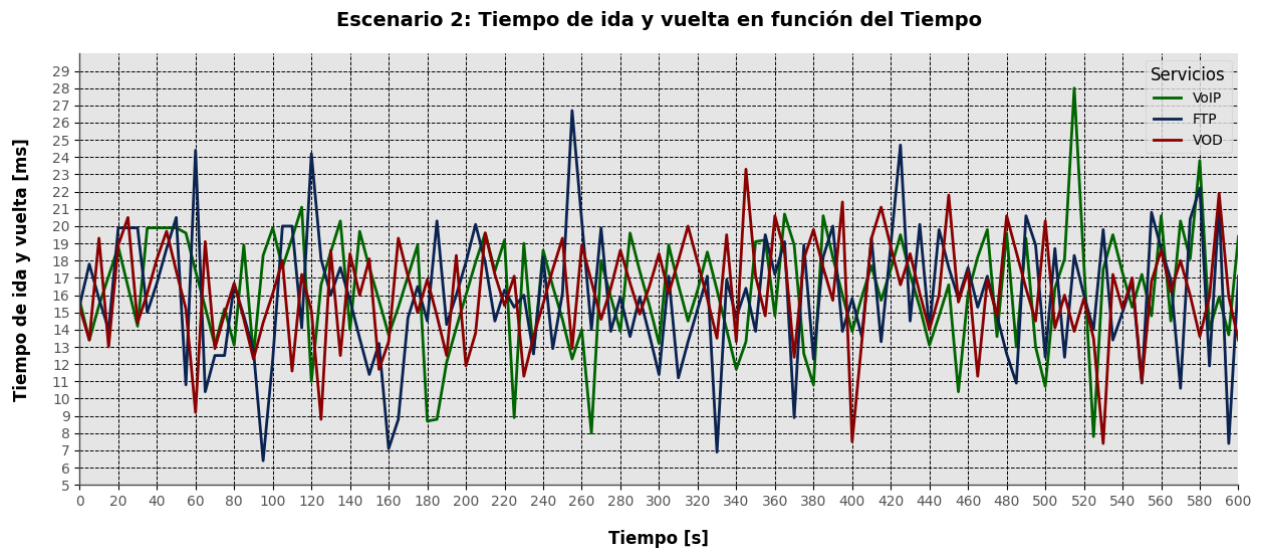
- **Tiempo de ida y vuelta (RTT)**

Los valores de RTT fueron obtenidos mediante la herramienta Hping3 y se presentan en la **Figura 118**. Esta gráfica muestra que el control de tráfico implementado es eficiente, permitiendo que los servicios mantengan valores aceptables de RTT que fluctúan entre los 6 y

28 ms. Estos resultados aseguran una latencia adecuada para servicios sensibles al tiempo, como lo es el caso de VoIP, al mismo tiempo que contribuyen al buen funcionamiento de VOD y FTP.

**Figura 118**

*Evaluación de RTT en escenario 2, con QoS.*

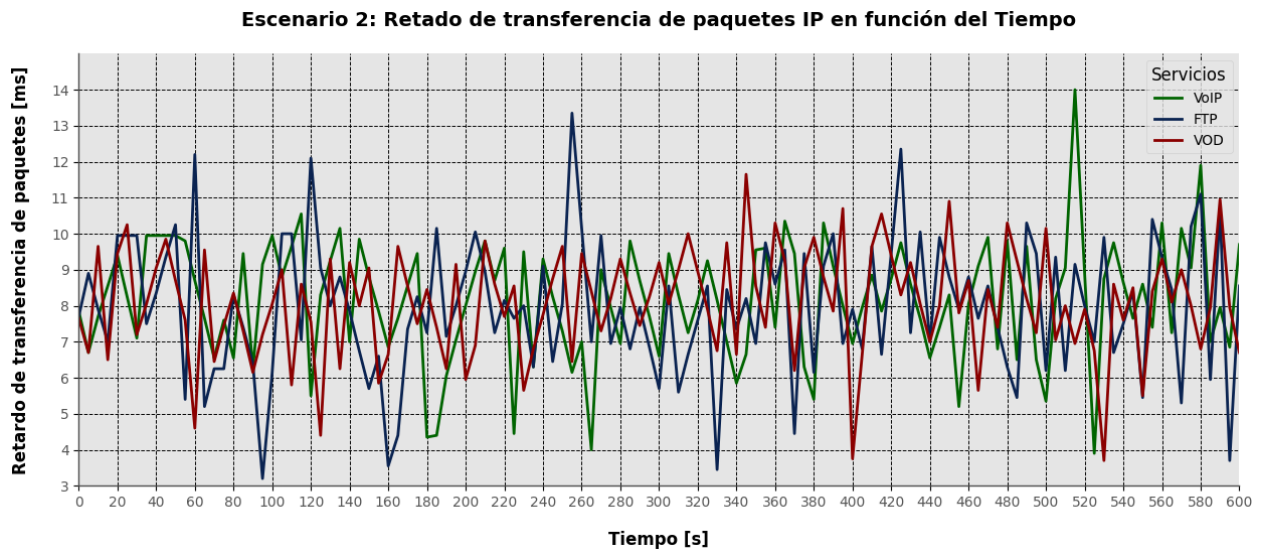


- **Retardo de transferencia de paquetes (IPTD)**

A partir de los datos obtenidos en la sección anterior y aplicando la fórmula de aproximación de latencia, se calcularon los valores de IPTD, representados en la **Figura 119**. En esta gráfica es posible apreciar que este KPI se mantiene en un rango óptimo y guarda una relación proporcional con los valores de RTT observados en la **Figura 118**.

**Figura 119**

*Evaluación de IPTD en escenario 2, con QoS.*



#### **4.1.3.2.3. Escenario 3: Tráfico intenso**

En este escenario, el MTU se configura en 45,000 bytes, lo que incrementa significativamente la carga en la red debido a la fragmentación necesaria para transmitir paquetes de gran tamaño. Esta configuración permite evaluar el desempeño de las políticas de QoS bajo condiciones de tráfico intenso. A continuación, se presenta el análisis de los cinco KPIs establecidos.

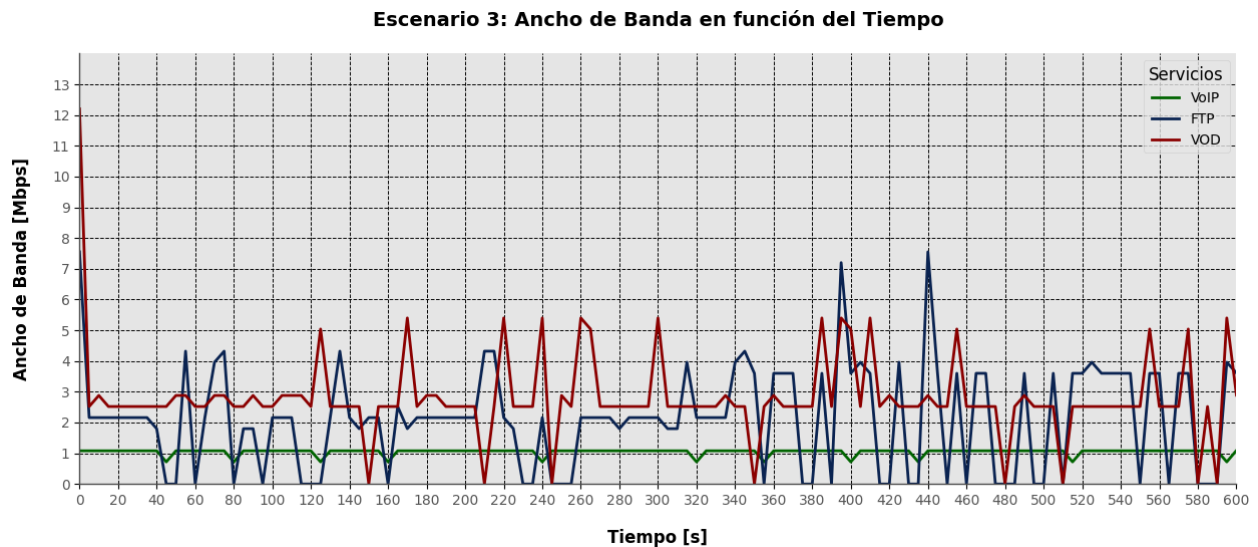
- **Ancho de Banda (AB)**

La **Figura 120** ilustra cómo, a pesar del incremento en el tamaño de los paquetes, las políticas de QoS permitieron que el servicio VoIP mantuviera un ancho de banda estable durante la evaluación. En contraste, los servicios VOD y FTP presentan una reducción significativa en el uso de ancho de banda debido a la fragmentación de paquetes y el nivel de priorización establecido.



**Figura 120**

*Evaluación de Ancho de banda en escenario 3, con QoS.*

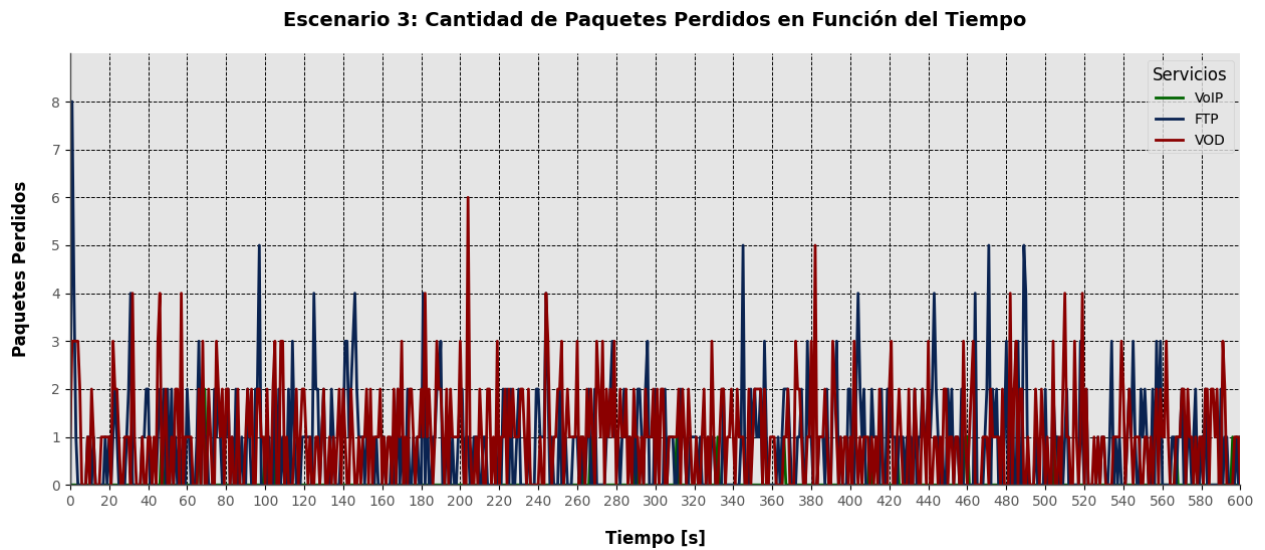


- **Tasa de paquetes perdidos (IPRL)**

La **Figura 121** muestra que la cantidad de paquetes perdidos se redujo en comparación con el escenario sin QoS (ver **Figura 93**). Según los datos recopilados, el servicio VoIP experimenta pérdidas a partir del segundo 47; no obstante, estas son mínimas, oscilando entre uno y dos paquetes perdidos. Por su parte, los servicios VOD y FTP presentan picos de pérdida más frecuentes y distribuidos de manera irregular a lo largo del tiempo. Sin embargo, estas pérdidas se mantienen dentro de un rango controlado, sin superar los 8 paquetes perdidos.

**Figura 121**

*Evaluación de paquetes perdidos en escenario 3, con QoS.*



De la evaluación con las herramientas iPerf3 y Wireshark, se obtiene la cantidad de paquetes transmitidos y perdidos, como lo evidencian las **Figuras 122, 123 y 124**. Con esta información se calcula el IPLR de los servicios, como se indica en la **Tabla 37**. Donde se visualiza que el servicio VOD obtuvo la menor tasa de paquetes perdidos en este escenario.

**Figura 122**

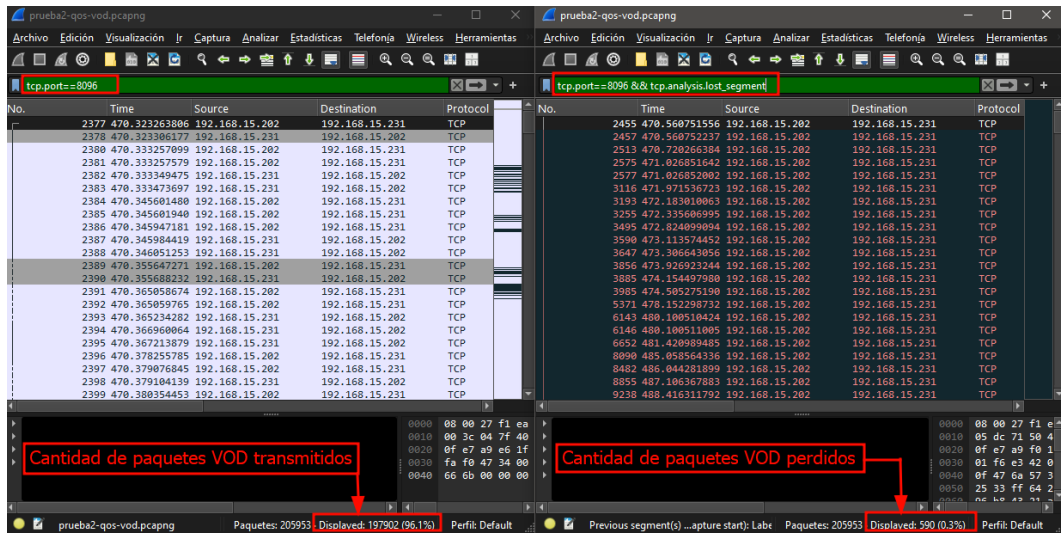
*Cantidad de paquetes perdidos y enviados para servicio VoIP.*

```

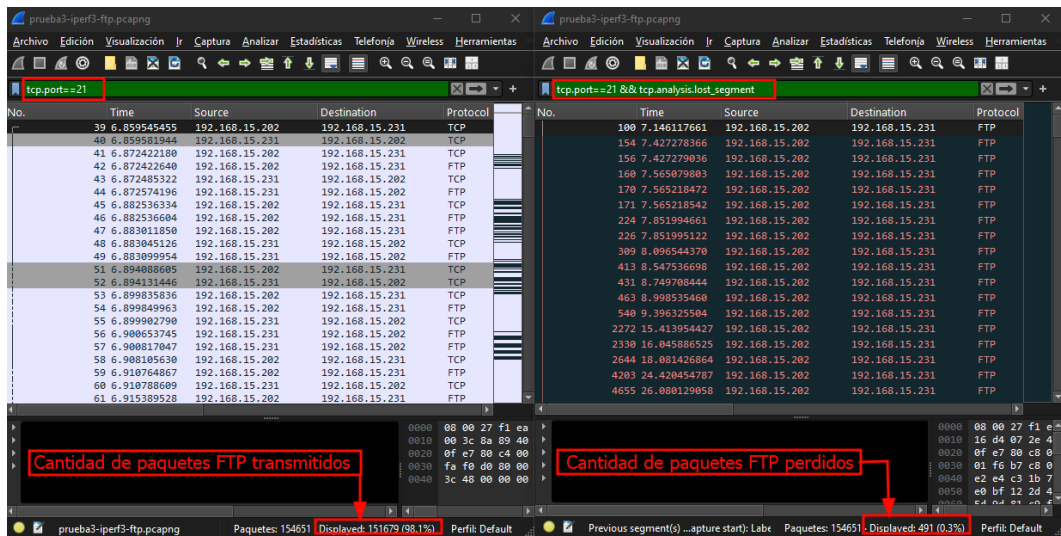
[ 5] 594.00-595.00 sec  132 KBytes  1.08 Mbits/sec  8.212 ms  0/3 (0%)
[ 5] 595.00-596.00 sec  132 KBytes  1.08 Mbits/sec  7.102 ms  0/3 (0%)
[ 5] 596.00-597.00 sec  87.9 KBytes  720 Kbits/sec  6.567 ms  1/3 (33%)
[ 5] 597.00-598.00 sec  132 KBytes  1.08 Mbits/sec  5.860 ms  0/3 (0%)
[ 5] 598.00-599.00 sec  132 KBytes  1.08 Mbits/sec  5.113 ms  0/3 (0%)
[ 5] 599.00-600.00 sec  132 KBytes  1.08 Mbits/sec  5.760 ms  0/3 (0%)
[ 5] 600.00-600.06 sec  0.00 Bytes  0.00 bits/sec  5.760 ms  0/0 (0%)
-----
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 5]  0.00-600.06 sec  0.00 Bytes  0.00 bits/sec  5.760 ms  25/1748 (1.4%)
-----

```

**Figura 123**  
Cantidad de paquetes perdidos y enviados para servicio VOD.



**Figura 124**  
Cantidad de paquetes perdidos y enviados para servicio FTP.



**Tabla 35**  
Datos de IPLR obtenidos en el Escenario 3, con QoS.

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	25	1748	0,0143
VOD	590	197902	0,0029
FTP	491	151679	0,0032

- **Disponibilidad de red**

La **Tabla 38** muestra que todos los servicios lograron mantenerse dentro de los límites de disponibilidad establecidos por la recomendación ITU-T Y.1540. Lo que indica que la red funcionó de manera más eficiente con las configuraciones de QoS establecidas.

**Tabla 36**

*Disponibilidad de red en Escenario 3, con QoS.*

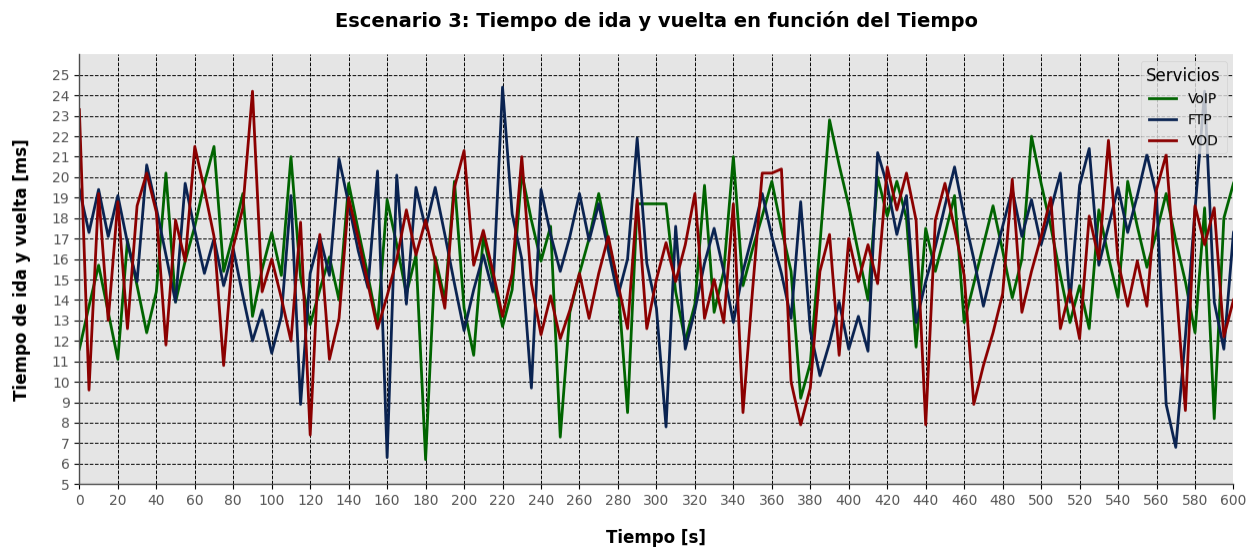
Servicio	Expresión Disponibilidad $IPRL < c_1$	Disponibilidad
VoIP	$0,0143 < 0,20$	Si
VOD	$0,0029 < 0,20$	Si
FTP	$0,0032 < 0,20$	Si

- **Tiempo de ida y vuelta (RTT)**

La **Figura 125** demuestra que el RTT para todos los servicios se mantuvo dentro de un rango aceptable, aunque hubo algunas fluctuaciones debido al mayor tamaño de los paquetes. Sin embargo, los picos de RTT fueron significativamente menores que los observados en el escenario sin QoS (ver **Figura 97**), indicando una mejora en la gestión de los tiempos de espera de los paquetes.

**Figura 125**

*Evaluación de Tiempo de ida y vuelta en escenario 3, con QoS.*

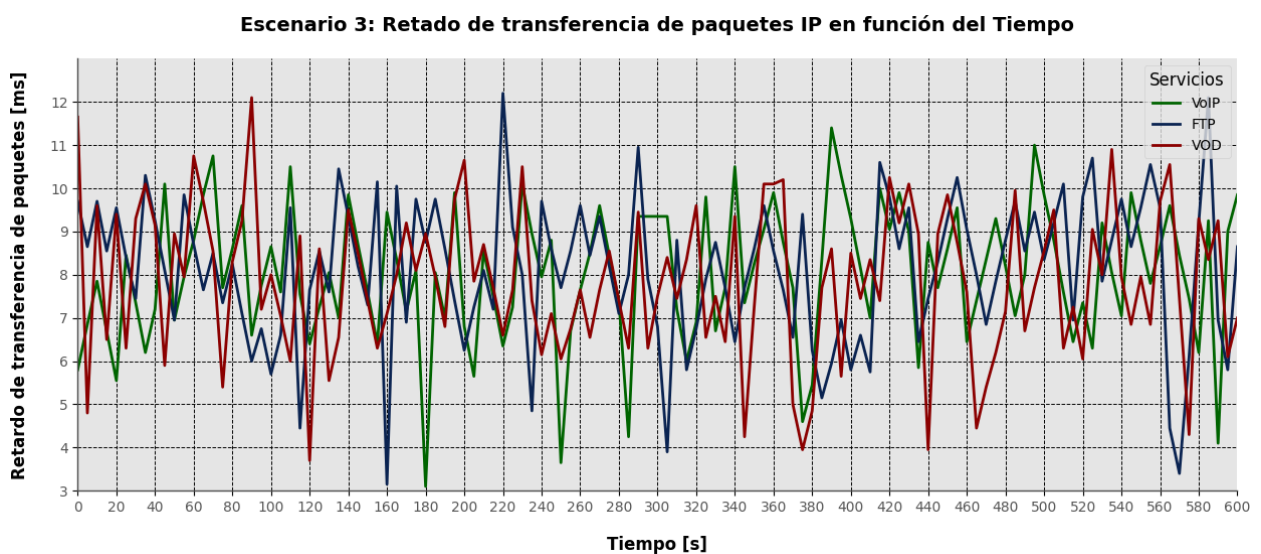


- **Retardo de transferencia de paquetes (IPTD)**

Los valores de IPTD, calculados a partir del RTT, se mantuvieron dentro de los límites aceptables, como se muestra en la **Figura 126**. De este modo se establece que las configuraciones de QoS permitieron un manejo eficiente del tráfico, minimizando los retardos incluso en condiciones de mayor carga de tráfico.

**Figura 126**

*Evaluación de IPTD en escenario 3, con QoS.*



#### 4.1.3.2.4. Escenario 4: Sobrecarga de red

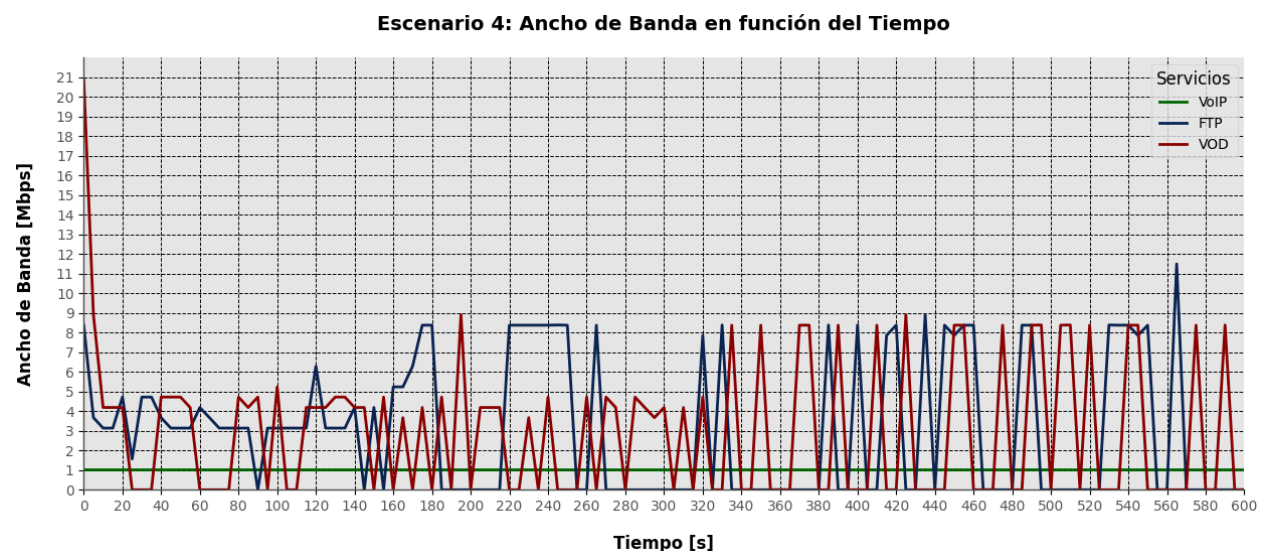
En este escenario, se ha configurado un valor de MTU de 65507 bytes, lo cual genera una sobrecarga de tráfico considerable en comparación con los tres escenarios anteriores. El objetivo es evaluar cómo se comporta la red bajo condiciones de alto tráfico. A continuación, se presenta la evaluación de los KPIs establecidos.

- **Ancho de Banda (AB)**

La evaluación del ancho de banda para este escenario se presenta en la **Figura 127**. En la gráfica, se observa que el servicio de VoIP mantiene un consumo de ancho de banda bajo y estable. En contraste, los servicios de VOD y FTP muestran una falta de estabilidad, con momentos críticos en los que el ancho de banda se reduce a cero bits por segundo, incluso con la configuración de QoS aplicada. Esto se debe a la sobrecarga de tráfico en la red, mismo que puede ser afectado por los limitados recursos de hardware de los dispositivos de red como es el caso de los puntos de acceso y el controlador SDN.

**Figura 127**

*Evaluación de Ancho de banda en escenario 4, con QoS.*

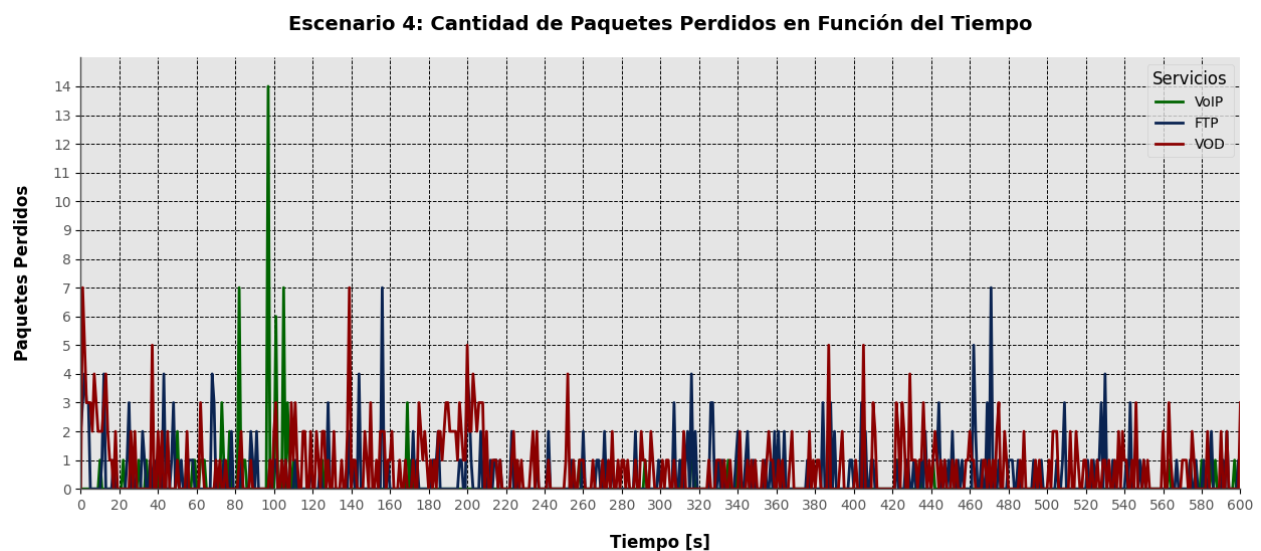


- **Tasa de paquetes perdidos (IPRL)**

Los datos relacionados con la cantidad de paquetes perdidos en este escenario se presentan en la **Figura 128**. En la gráfica se puede apreciar que, aunque el servicio VoIP experimenta algunos picos significativos de pérdida, como el que ocurre cerca del segundo 100 donde se supera la cifra de 14 paquetes, en general las pérdidas son aisladas. Por otro lado, los servicios VOD y FTP muestran pérdidas distribuidas a lo largo del tiempo, estas se mantienen bajo control y no superan los 8 paquetes. Esto demuestra la eficacia de las políticas de QoS para reducir la pérdida de paquetes y garantizar la estabilidad en la transmisión de datos.

**Figura 128**

*Evaluación de paquetes perdidos en escenario 4, con QoS.*



Utilizando las herramientas iPerf3 y Wireshark, se registró la cantidad de paquetes transmitidos y perdidos durante la evaluación. Estos resultados se ilustran en las **Figuras 129**, **130** y **131**. A partir de esta información, se calculó el valor de IPLR para los distintos servicios, como se muestra en la **Tabla 39**. En esta tabla se destaca que el servicio FTP presentó la menor tasa de pérdida de paquetes en este escenario.



**Figura 129**  
Cantidad de paquetes perdidos y enviados para servicio VoIP.

[ 5 ]	593.00-594.00 sec	128 KBytes	1.05 Mbits/sec	13.340 ms	0/2 (0%)
[ 5 ]	594.00-595.00 sec	128 KBytes	1.05 Mbits/sec	12.167 ms	0/2 (0%)
[ 5 ]	595.00-596.00 sec	128 KBytes	1.05 Mbits/sec	11.898 ms	0/2 (0%)
[ 5 ]	596.00-597.00 sec	64.0 KBytes	524 Kbits/sec	12.746 ms	0/1 (0%)
[ 5 ]	597.00-598.00 sec	128 KBytes	1.05 Mbits/sec	11.297 ms	1/3 (33%)
[ 5 ]	598.00-599.00 sec	128 KBytes	1.05 Mbits/sec	10.303 ms	0/2 (0%)
[ 5 ]	599.00-600.00 sec	128 KBytes	1.05 Mbits/sec	10.629 ms	0/2 (0%)
[ 5 ]	600.00-600.06 sec	0.00 Bytes	0.00 bits/sec	10.629 ms	0/0 (0%)

[ ID ]	Interval	Transfer	Bandwidth	Jitter	Lost/Total Datagrams
[ 5 ]	0.00-600.06 sec	0.00 Bytes	0.00 bits/sec	10.629 ms	83/1201 (6.9%)

**Figura 130**  
Cantidad de paquetes perdidos y enviados para servicio VOD.

**Figura 131**  
Cantidad de paquetes perdidos y enviados para servicio FTP.



**Tabla 37**

*Datos de IPLR obtenidos en el Escenario 4, con QoS.*

Servicio	Paquetes perdidos	Paquetes Transmitidos	IPLR
VoIP	83	1201	0,0691
VOD	413	210229	0,0019
FTP	261	245266	0,0010

- **Disponibilidad de red**

La **Tabla 18** muestra que todos los servicios cumplieron con los requisitos de disponibilidad establecidos, asegurando que la red permaneciera funcional incluso bajo condiciones de sobrecarga de tráfico.

**Tabla 38**

*Disponibilidad de red en Escenario 4, con QoS.*

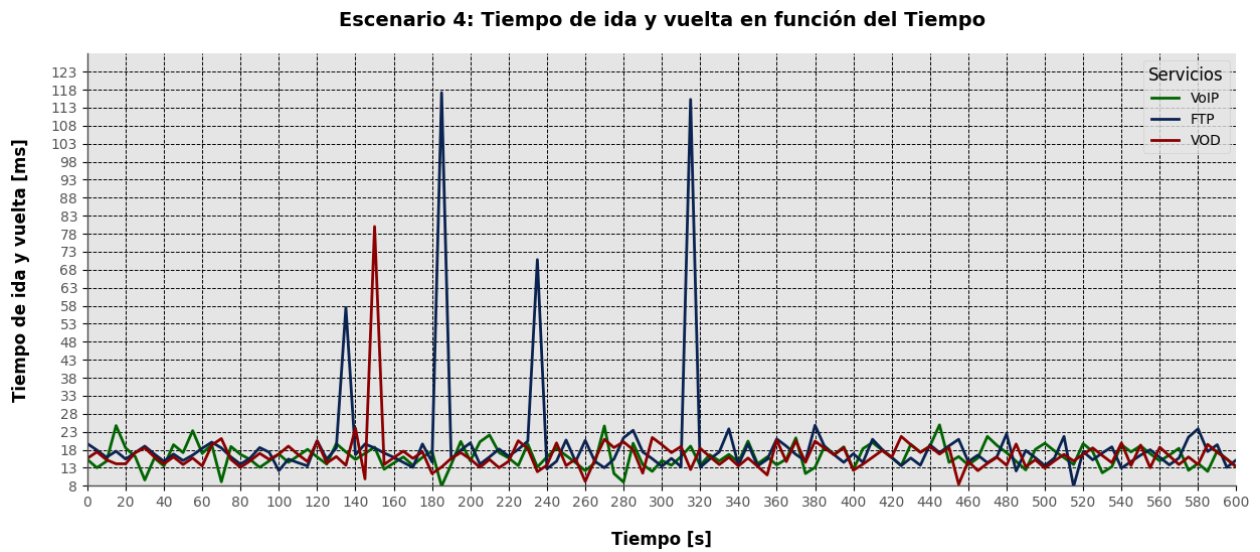
Servicio	Expresión Disponibilidad $IPLR < c_1$	Disponibilidad
VoIP	$0,0691 < 0,20$	Si
VOD	$0,0019 < 0,20$	Si
FTP	$0,0010 < 0,20$	Si

- **Tiempo de ida y vuelta (RTT)**

La **Figura 132** muestra que el RTT en este escenario fue relativamente estable, con algunos picos en los servicios VOD y FTP debido a la congestión, donde FTP es el más afectado debido a la baja prioridad asignada. Sin embargo, QoS ayudó a mitigar los grandes picos y a mantener los tiempos de respuesta dentro de límites aceptables.

**Figura 132**

*Evaluación de Tiempo de ida y vuelta en escenario 4, con QoS.*



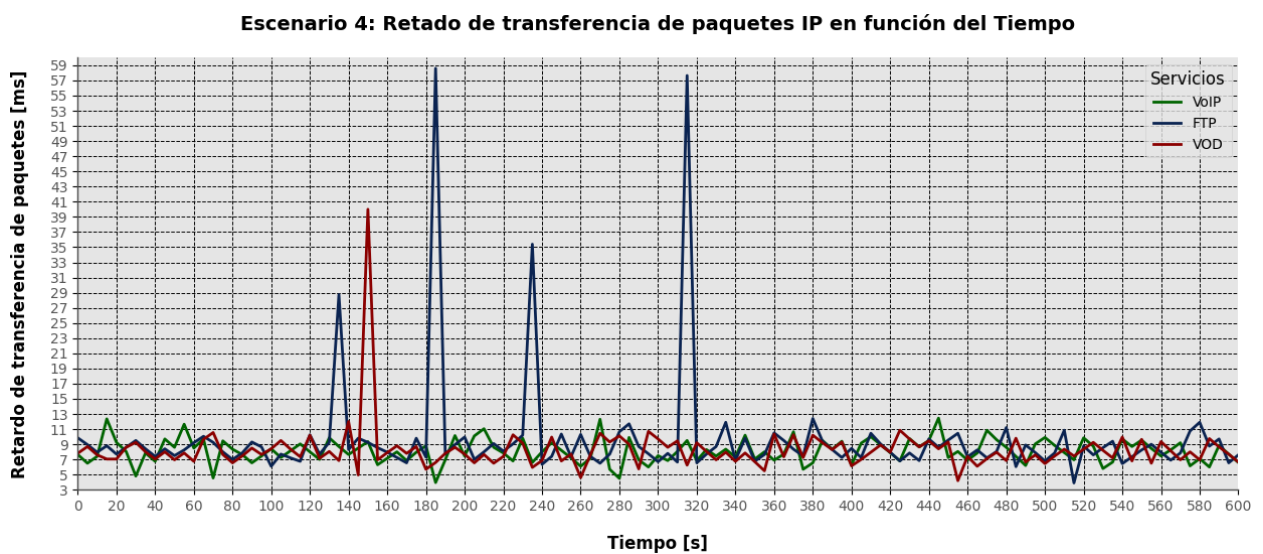
- **Retardo de transferencia de paquetes (IPTD)**

Al igual que en los escenarios anteriores, los datos de IPTD son proporcionales a los valores de RTT debido a la aproximación empleada. En este escenario, los valores de IPTD se mantuvieron dentro de los límites adecuados, con fluctuaciones mínimas, como se ilustra en la

**Figura 133.**

**Figura 133**

*Evaluación de IPTD en escenario 4, con QoS.*



#### ***4.1.4. Resumen y comparación de pruebas de rendimiento***

En este trabajo de titulación se realizaron diversas pruebas en una red SDWN con el propósito de analizar y comparar su rendimiento en condiciones operativas, tanto sin mecanismos de QoS como bajo la implementación de estos. Para ello, se siguieron las directrices establecidas en la Recomendación ITU-T Y.1540, las cuales abordan aspectos como la velocidad, exactitud y seguridad de funcionamiento. Los escenarios de prueba se establecieron en función de diferentes condiciones de tráfico, determinadas por el valor de MTU de los paquetes. Se evaluó el rendimiento de la red mediante cinco indicadores clave de rendimiento (KPIs): Ancho de Banda (AB), Tasa de Pérdida de Paquetes (IPLR), Disponibilidad de red, Tiempo de ida y vuelta (RTT) y Retardo de Transferencia de Paquetes (IPTD).

Por otro lado, el tráfico simulado y la medición de los KPIs para cada escenario se realizó con herramientas basadas en software como iPerf3 y Hping3. Adicionalmente, se utilizó el sniffer Wireshark para capturar el tráfico de red y obtener estadísticas sobre la cantidad de paquetes perdidos de los servicios que funcionan bajo TCP, ya que las herramientas anteriores no proporcionaban dichos datos. De este modo, se obtienen datos referentes a los cinco KPIs definidos, lo que permite conocer el rendimiento de la red.

De manera general, los resultados de las pruebas efectuadas en la red antes de aplicar QoS muestran fluctuaciones significativas en el rendimiento de los servicios debido a la competencia de recursos en la red. VoIP experimentó pérdidas de paquetes excesivas en los escenarios 2, 3 y 4, por lo cual también presentó indisponibilidad en dichos escenarios. Los servicios VOD y FTP tuvieron picos de uso que llegaron a saturar los recursos disponibles, lo que afectó a la estabilidad de la red.

Tras la implementación de QoS, los resultados mejoraron sustancialmente. El servicio VoIP, con alta prioridad, mostró una reducción significativa de la tasa de pérdida de paquetes, garantizando la estabilidad en su funcionamiento. Los servicios VOD y FTP experimentaron mejoras en el manejo del ancho de banda, sin comprometer el rendimiento del VoIP. La disponibilidad de la red también mejoró, ya que los valores de IPLR se mantuvieron por debajo del umbral de 0,20 establecido por la ITU-T Y.1540, garantizando que los servicios permanecieran disponibles durante las pruebas.

A continuación, se presentan los resultados de los cinco KPIs obtenidos durante las pruebas, comparando los escenarios antes y después de aplicar QoS. Estos datos ofrecen una visión clara y detallada del impacto que tienen las configuraciones de QoS en el rendimiento de la red SDWN.

- **Ancho de banda (AB)**

Los resultados del ancho de banda obtenidos en las pruebas se presentan en la **Tabla 39**. Evidenciando que, sin QoS, el ancho de banda del servicio VoIP se ve afectado por la competencia de recursos, donde los servicios VOD y FTP alcanzan altos valores, mientras que VoIP se mantiene en niveles bajos, llegando incluso a 0,66 Mbps. En cambio, tras la aplicación QoS, el servicios VoIP muestra un ancho de banda estable de 1,05 Mbps en todos los escenarios, debido a su alta prioridad en la red. Por otro lado, VOD y FTP ajustaron su uso permitiendo una distribución más eficiente del ancho de banda.

La **Figura 134** presenta las gráficas de las evaluaciones realizadas en los cuatro escenarios, tanto antes como después de aplicar QoS en la red. En la figura, la columna izquierda recopila las gráficas de la red sin QoS, las cuales muestran que a medida que aumenta el tráfico, los servicios experimentan mayores fluctuaciones. En el escenario 3, se observa una disminución considerable del ancho de banda para todos los servicios, causada por la fragmentación de

paquetes necesarios para la transmisión con grandes valores de MTU y las retransmisiones de servicios bajo TCP. En contraste, la columna derecha, que muestra las gráficas de la red con QoS, evidencia una estabilidad completa para VoIP. Los servicios VOD y FTP presentan fluctuaciones que aumentan con el incremento del tráfico en la red, siendo el servicio FTP el más afectado debido a su menor prioridad en la red.

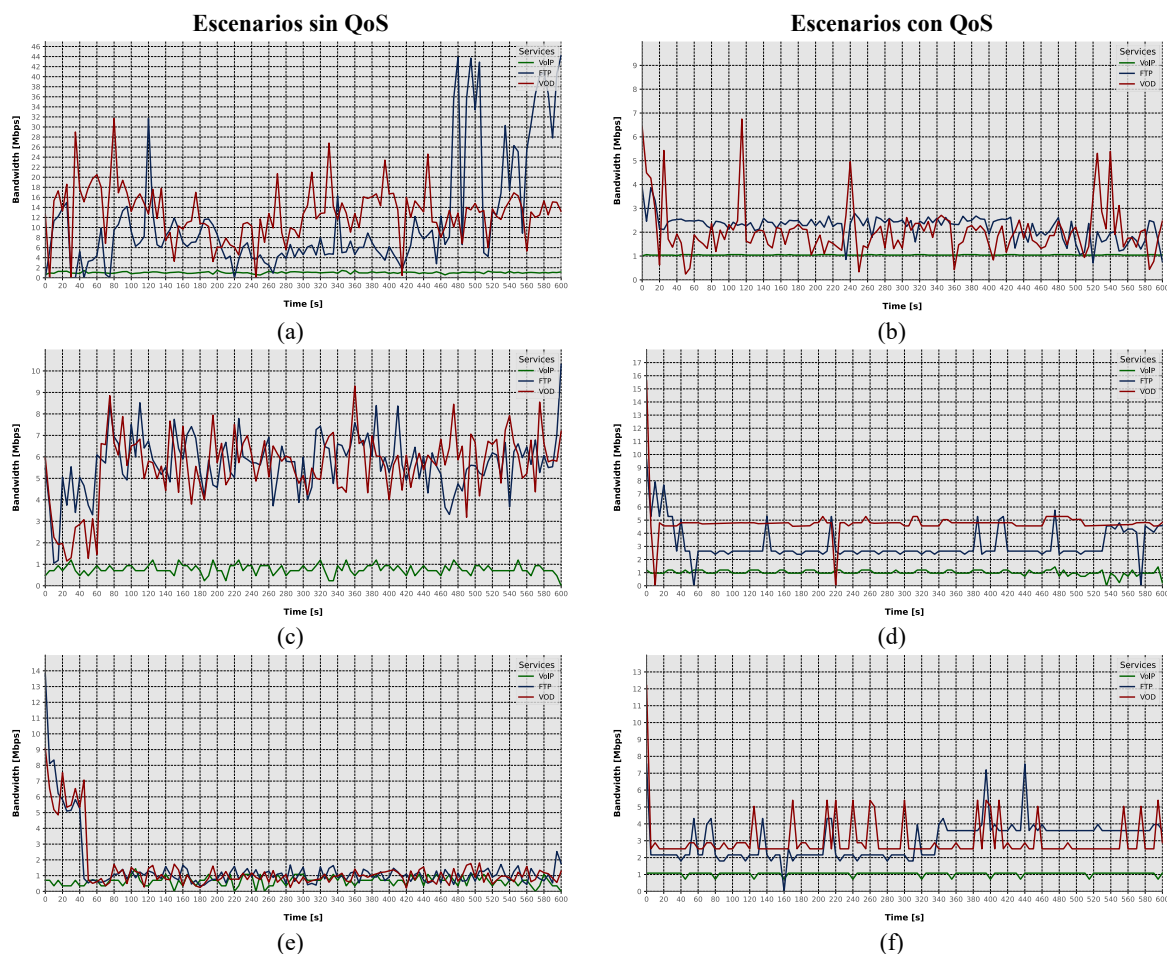
**Tabla 39**

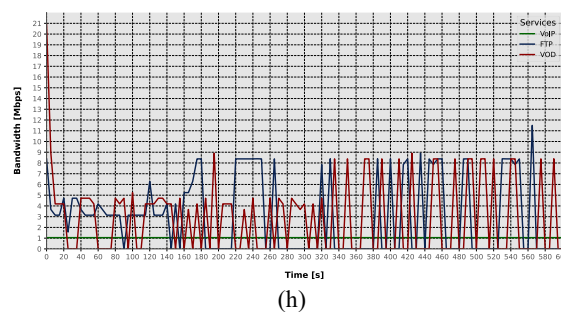
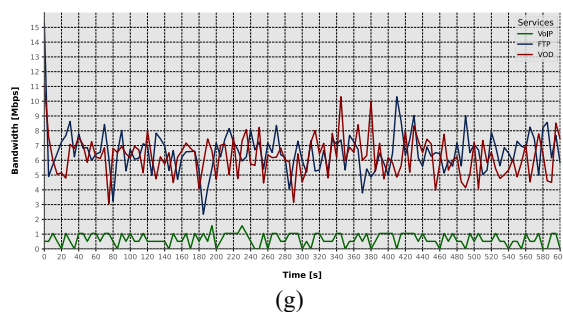
*Ancho de banda promedio en escenarios de pruebas con y sin QoS.*

Escenarios	MTU [Bytes]	AB promedio sin QoS			AB promedio con QoS		
		VoIP [Mbps]	VOD [Mbps]	FTP [Mbps]	VoIP [Mbps]	VOD [Mbps]	FTP [Mbps]
Escenario 1	1500	1,03	12,90	11,31	1,05	2,05	2,23
Escenario 2	30000	0,76	5,59	5,72	1,05	4,78	3,13
Escenario 3	45000	0,66	1,38	1,46	1,05	2,83	2,14
Escenario 4	65507	0,66	6,27	6,57	1,05	2,94	3,05

**Figura 134**

*Comparación de AB en escenarios evaluados antes y después de aplicar QoS.*





*Nota.* Los gráficos se distribuyen de la siguiente manera: **a** y **b** Escenario 1; **c** y **d** Escenario 2; **e** y **f** Escenario 3; **g** y **h** Escenario 4.

- **Tasa de paquetes perdidos (IPLR)**

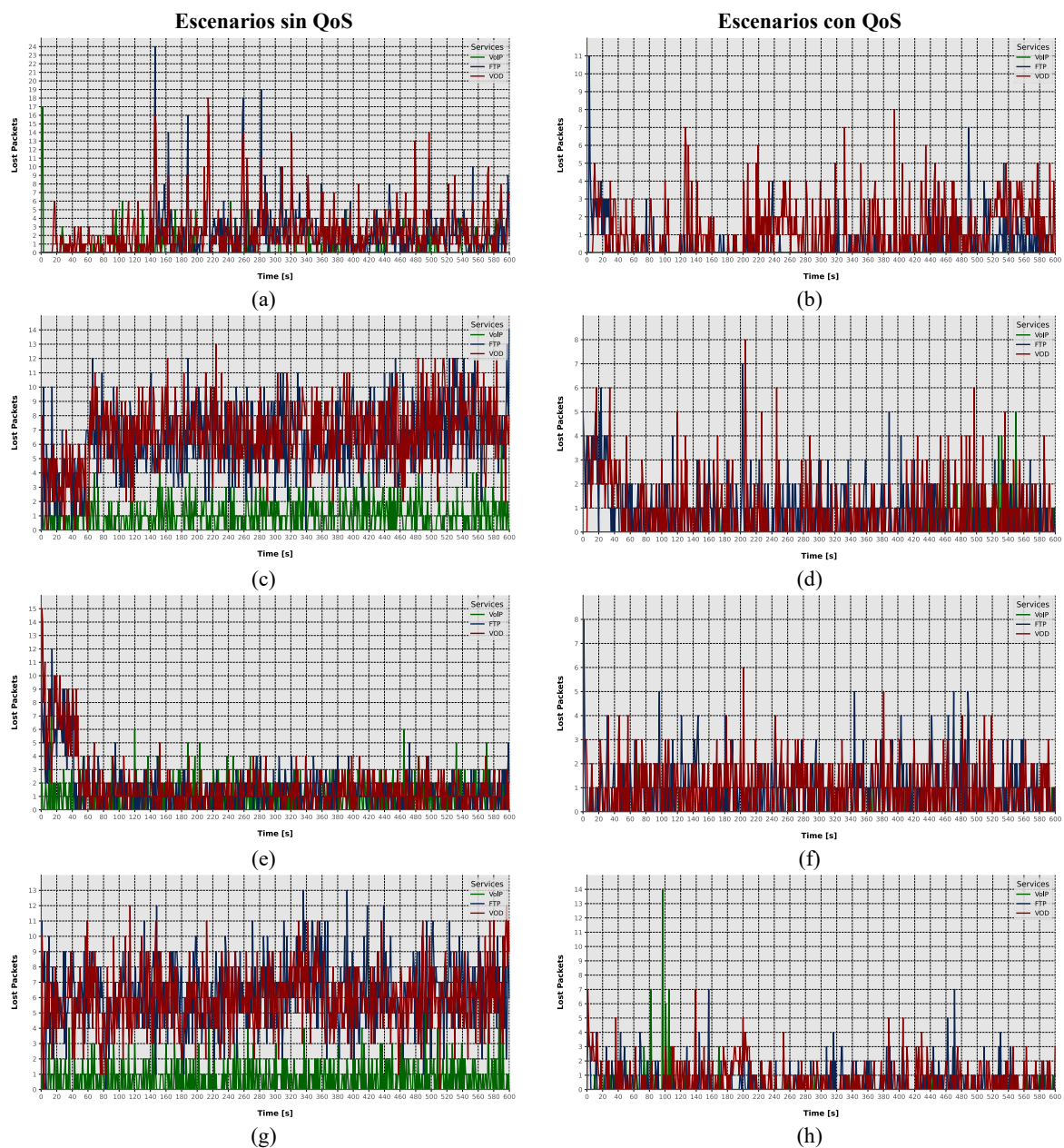
La **Tabla 40** presenta el IPLR promedio obtenido en las pruebas realizadas. Los resultados muestran que, en ausencia de QoS, el servicio de VoIP experimentó una alta tasa de pérdida de paquetes en los escenarios 2, 3 y 4, alcanzando valores de hasta 0.388, lo cual provocó en la indisponibilidad del servicio. Sin embargo, tras la implementación de QoS, se observó una disminución significativa en la tasa de pérdida de paquetes para todos los servicios. En el caso de VoIP, no se registraron pérdidas en el escenario 1, y aunque hay pérdidas en los escenarios 2, 3 y 4, estas no comprometieron la disponibilidad de este. Por su parte, VOD y FTP también experimentaron mejoras notables, manteniendo la tasa de pérdida por debajo de 0.005 en la mayoría de los escenarios.

La **Figura 135** presenta las gráficas correspondientes a los cuatro escenarios evaluados, comparando el desempeño de la red antes y después de implementar QoS. Los datos reflejan que, en la red sin QoS, los servicios VOD y FTP alcanzaron picos promedio de hasta 16 paquetes perdidos. En contraste, tras la implementación de QoS, las gráficas evidencian una mejora significativa, con picos promedio de alrededor de 10 paquetes perdidos en la mayoría de los intervalos para ambos servicios. En el caso de VoIP, las pruebas sin QoS evidenciaron pérdidas recurrentes y constantes a lo largo de todos los instantes evaluados. En cambio, con QoS, las pérdidas dejaron de ser consecutivas, mejorando la estabilidad del servicio en todos los escenarios analizados.

**Tabla 40**  
Tasa de paquetes perdidos en escenarios de pruebas con y sin QoS.

Escenarios	IPLR promedio sin QoS			IPLR promedio con QoS		
	VoIP	VOD	FTP	VoIP	VOD	FTP
Escenario 1	0,0043	0,0043	0,0043	0	0,0050	0,0013
Escenario 2	0,2430	0,0149	0,0158	0,0293	0,0034	0,0023
Escenario 3	0,3472	0,0106	0,0101	0,0143	0,0029	0,0032
Escenario 4	0,3880	0,0143	0,0135	0,0691	0,0019	0,0010

**Figura 135**  
Comparación de paquetes perdidos en escenarios evaluados sin y con QoS.



*Nota.* Los gráficos se distribuyen de la siguiente manera: **a** y **b** Escenario 1; **c** y **d** Escenario 2; **e** y **f** Escenario 3; **g** y **h** Escenario 4.

- **Disponibilidad de red**

La disponibilidad de los servicios en la red fue evaluada utilizando la **Ec. 2** cuya expresión relaciona  $IPRL < c_1$ , donde  $c_1 = 0,20$  corresponde al valor de referencia establecido por la recomendación ITU-T Y.1540 (2019). En este contexto, la **Tabla 41** recopila los datos de disponibilidad en los escenarios analizados. Los resultados muestran que, sin de QoS, el servicio VoIP se presentó indisponibilidad en los escenarios 2, 3 y 4 debido a la saturación de recursos. Sin embargo, tras la implementación de QoS, todos los servicios se mantuvieron disponibles en todos los escenarios evaluados, lo que evidencia una mejora significativa en la estabilidad y accesibilidad de la red.

**Tabla 41**

*Disponibilidad de servicios en escenarios de pruebas con y sin QoS.*

Escenarios	Disponibilidad sin QoS			Disponibilidad con QoS		
	VoIP	VOD	FTP	VoIP	VOD	FTP
<b>Escenario 1</b>	Si	Si	Si	Si	Si	Si
<b>Escenario 2</b>	No	Si	Si	Si	Si	Si
<b>Escenario 3</b>	No	Si	Si	Si	Si	Si
<b>Escenario 4</b>	No	Si	Si	Si	Si	Si

- **Tiempo de ida y vuelta (RTT)**

Los valores de RTT obtenidos en la evaluación de la red se presentan en la **Tabla 42**. Los resultados muestran que, en los escenarios 1, 2 y 3 sin QoS, los valores de RTT se mantiene moderados. Sin embargo, en el escenario 4, estos valores suben hasta alcanzar el valor promedio de 63,4 ms. Por otro lado, los escenarios con QoS evidencian una mayor estabilidad, incluso en el escenario 4, donde a pesar de la alta carga de red, los valores promedio para todos los servicios se encuentran alrededor de los 16 ms.

La **Figura 136** compara las gráficas obtenidas de las pruebas en la red. De estas gráficas se evidencia que, en los escenarios sin QoS, el RTT experimenta fluctuaciones significativas

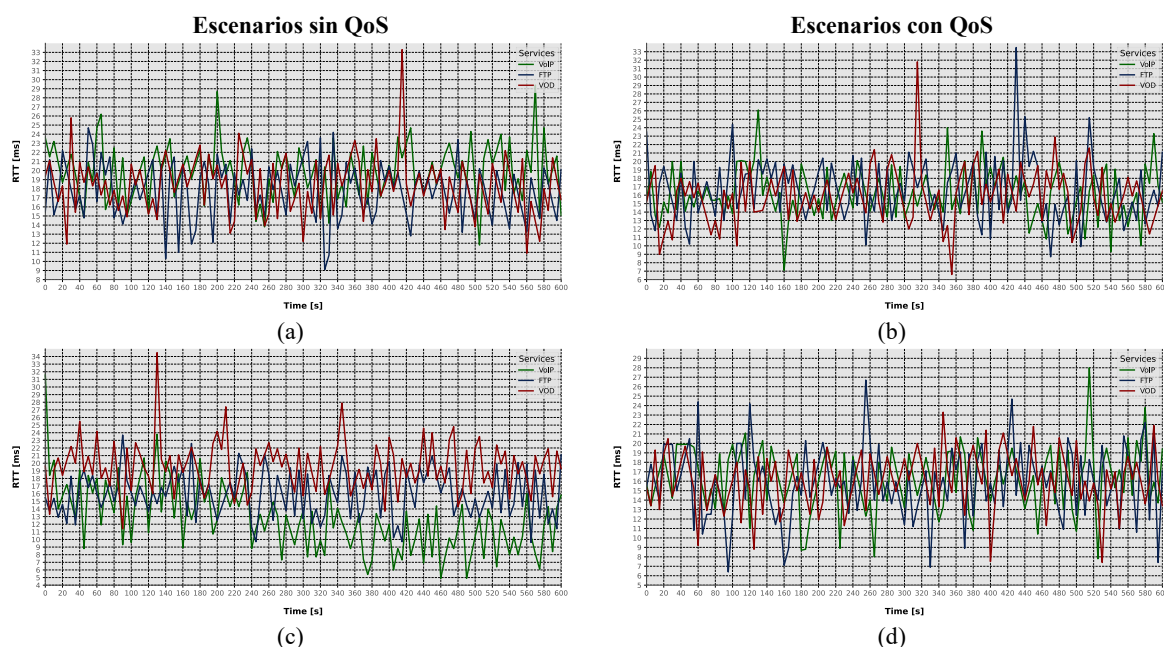


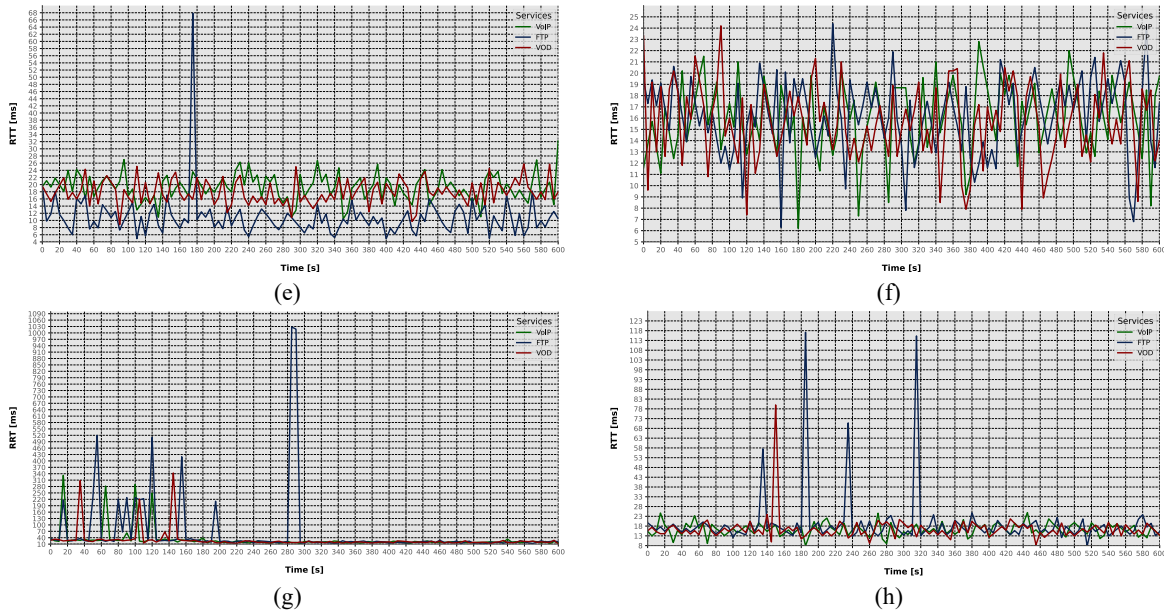
especialmente en aquellos escenarios con mayor demanda de tráfico como los escenarios 3 y 4, que utilizan MTU de 30000 y 65507 bytes respectivamente. En estos casos, se observan picos elevados de hasta 1,030 ms, particularmente en el servicio FTP. Por otro lado, con la implementación de QoS, los valores de RTT se mantienen más estables en los tres primeros escenarios. Sin embargo, en el cuarto escenario, persisten fluctuaciones considerables en los servicios VOD y FTP, aunque el servicio VoIP permanece sin afectaciones.

**Tabla 42**  
*RTT promedio en escenarios de pruebas con y sin QoS.*

Escenarios	RTT promedio sin QoS			RTT promedio con QoS		
	VoIP [ms]	VOD [ms]	FTP [ms]	VoIP [ms]	VOD [ms]	FTP [ms]
<b>Escenario 1</b>	19,4	18,5	17,6	16,2	15,9	16,6
<b>Escenario 2</b>	12,5	19,2	16,1	16,3	16,1	15,9
<b>Escenario 3</b>	19,1	18,8	10,2	16	15,7	16,2
<b>Escenario 4</b>	33,7	31	63,4	16,2	16,5	19,2

**Figura 136**  
*Comparación de RTT en escenarios evaluados antes y después de QoS.*





**Nota.** Los gráficos se distribuyen de la siguiente manera: **a** y **b** Escenario 1; **c** y **d** Escenario 2; **e** y **f** Escenario 3; **g** y **h** Escenario 4.

- **Retardo de transferencia de paquetes (IPTD)**

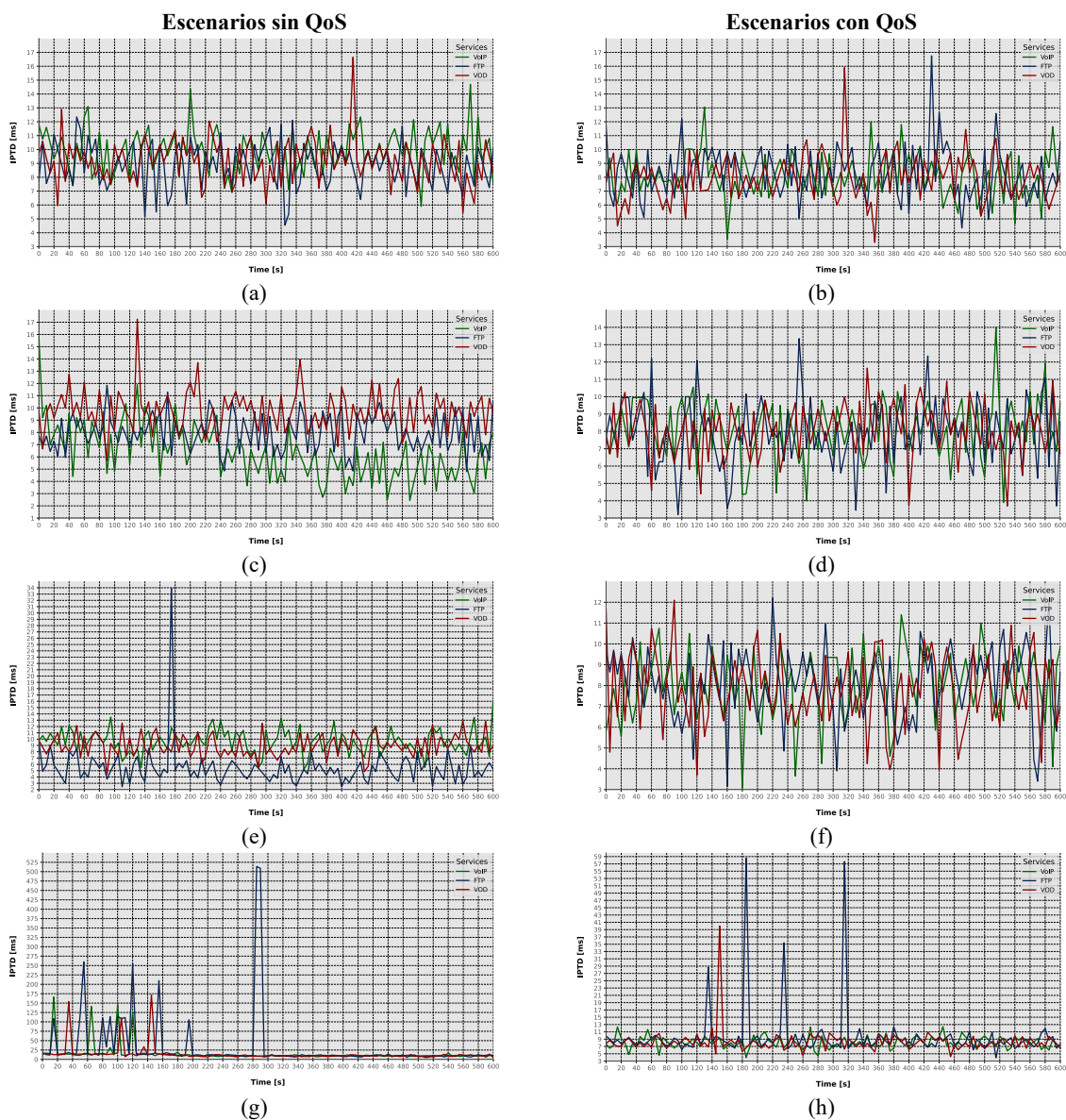
Los valores del IPTD, también conocido como latencia, fueron calculados utilizando la aproximación  $latencia = \frac{RTT}{2}$ . De este modo, se emplea los datos de RTT recopilados con Hping3. Así, la Tabla 43 presenta los valores promedio de IPTD para los diferentes servicios evaluados. De este datos es posible analizar que, en los escenarios sin QoS, el IPTD mostró incrementos significativos, destacando el escenario 4, donde VoIP alcanzó un promedio de 16.9 ms y FTP llegó hasta 31.7 ms. En contraste, los datos obtenidos tras implementar QoS evidencian una reducción considerable del retardo promedio en todos los servicios: VoIP se estabilizó en 8.1 ms, mientras que los servicios VOD y FTP se mantuvieron dentro de rangos óptimos, asegurando un desempeño adecuado.

La **Figura 137** recopila las gráficas correspondientes a los escenarios evaluados antes y después de aplicar QoS. En estas gráficas se observa que los valores de IPTD presentan una tendencia proporcional a los de RTT, debido a la aproximación utilizada en los cálculos. Además, todos los servicios muestran una mayor estabilidad en los escenarios con QoS, con valores que oscilan en torno a los 8 ms en la mayoría de los casos.

**Tabla 43**  
*IPTD promedio en escenarios de pruebas con y sin QoS.*

Escenarios	IPTD promedio sin QoS			IPTD promedio con QoS		
	VoIP [ms]	VOD [ms]	FTP [ms]	VoIP [ms]	VOD [ms]	FTP [ms]
<b>Escenario 1</b>	9,7	9,3	8,8	8,1	7,9	8,3
<b>Escenario 2</b>	6,3	9,6	8,1	8,2	8,1	7,9
<b>Escenario 3</b>	9,6	9,4	5,1	8	7,9	8,1
<b>Escenario 4</b>	16,9	15,5	31,7	8,1	8,3	9,6

**Figura 137**  
*Comparación de IPTD en escenarios evaluados antes y después de QoS.*



**Nota.** Los gráficos se distribuyen de la siguiente manera: **a** y **b** Escenario 1; **c** y **d** Escenario 2; **e** y **f** Escenario 3; **g** y **h** Escenario 4.

## 5. CONCLUSIONES

- Durante la revisión bibliográfica efectuada en el desarrollo de este trabajo de grado, se identificó que en entornos SDN es factible implementar una amplia variedad de mecanismos de QoS, que van desde los tradicionales como Diffserv, hasta métodos avanzados como los basados en tablas de medidores introducidos a partir de la versión 1.3 del protocolo OpenFlow.
- Se pudo establecer el prototipo SDWN donde se integró configuraciones de QoS basadas en el mecanismo DiffServ. Para ello se emplearon los recursos de hardware y software seleccionados, lo que permitió alcanzar los resultados esperados en cuanto a la clasificación de tráfico en función de la Recomendación ITU-T Y.1541, donde las funcionalidades de encolamiento, marcaje y priorización fueron aplicadas exitosamente. Estableciendo así una base sólida para la realización de pruebas de rendimiento en la red.
- En conformidad con las directrices definidas por la recomendación ITU-T Y.1540 se definieron escenarios de pruebas y parámetros de rendimiento (KPIs), alineados a los aspectos de velocidad, exactitud y seguridad de funcionamiento, que permitieron recopilar datos necesarios para evaluar el rendimiento de red tanto antes como después de aplicar las configuraciones de QoS, con el propósito de analizar su funcionalidad y determinar la eficiencia del mecanismo DiffServ implementado.
- Las pruebas de rendimiento realizadas en la red demostraron que, en ausencia de QoS, los servicios VOD y FTP experimentaron altos picos de uso que llegaron a saturar los recursos de red disponibles, afectando negativamente la estabilidad del servicio VoIP. En contraste, al implementar QoS, se logró una distribución eficiente de los recursos de red, garantizando la disponibilidad y el buen funcionamiento de todos los servicios, lo

cual demostró la efectividad del mecanismo DiffServ en la gestión y optimización del tráfico de red.

- La gestión de QoS en entornos SDN ofrece un enfoque centralizado que optimiza significativamente la configuración y administración de dispositivos de red, al delegar estas tareas al controlador principal. Además, los resultados obtenidos en las pruebas de RTT bajo condiciones de tráfico intensas, incluso sin QoS, evidenciaron valores relativamente bajos, destacando así la eficiencia de las arquitecturas SDN en términos de rendimiento y capacidad para manejar condiciones de tráfico intensas.

## **6. RECOMENDACIONES**

- Se sugiere explorar los mecanismos avanzados de QoS en SDN disponibles en las versiones más recientes y estables del protocolo OpenFlow, para identificar si se adapta a las necesidades específicas de la red y así poder alcanzar los objetivos de rendimiento deseados.
- Se recomienda aprovechar las soluciones de código abierto para implementar entornos SDN basados en la arquitectura OpenFlow la cual consta de un controlador, un switch OpenFlow y un canal seguro. Debido a que la ONF define al protocolo OpenFlow como la interfaz de comunicación estándar entre los planos de control y datos en la arquitectura SDN, por lo cual la comprensión de este protocolo resulta fundamental para el estudio y desarrollo de redes definidas por software.
- Es recomendable ampliar los escenarios de pruebas en entornos SDN, incorporando aplicaciones que exigen mayores requisitos de calidad como la transmisión de video UHD y los juegos en línea, para validar la adaptabilidad del sistema bajo las diferentes demandas actuales.
- Se recomienda probar soluciones SDN para el despliegue de arquitecturas de red, aprovechando su configuración centralizada. Característica que facilita una

escalabilidad eficiente al permitir la gestión y configuración centralizada desde un único controlador, eliminando la necesidad de ajustes manuales en múltiples dispositivos.

- Para trabajos futuros, se sugiere experimentar redes más complejas donde se utilice dos controladores red, asignando a cada controlador la gestión de una parte específica de la red. Esta división permitirá optimizar la administración y reducir la carga de trabajo en un único controlador. Para ello, cada instancia debe configurarse de manera que se comunique eficientemente con los switches OpenFlow correspondientes y con las interfaces REST necesarias para aplicar las políticas de QoS.

## 7. REFERENCIAS

- Aibin, M. (2024, March 18). *www.baeldung.com*. ¿Cómo Describir El Rendimiento de La Red? <https://www.baeldung.com/cs/bandwidth-packet-loss-latency-jitter>
- Alarcón, R. (2003). *Estudio e Implementación de Mecanismos de Calidad de Servicio sobre una Arquitectura de Servicios Diferenciados* [Universidad Politécnica de Cartagena]. <https://repositorio.upct.es/bitstream/handle/10317/184/pfc908.pdf?sequence=1&i>
- Albu-Salih, A. T. (2022). Performance Evaluation of Ryu Controller in Software Defined Networks. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, 14(1). <https://doi.org/10.29304/jqcm.2022.14.1.879>
- Alkahtani, A. M. S., Woodward, M. E., & Al-Begain, K. (2003). *An Overview of Quality of Service (QoS) and QoS Routing in Communication Networks*. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e970d7fceb60842591e4b7ed244868ee4b16127>
- AWS. (2023). *aws.amazon.com*. ¿Qué Es El RTT En Redes? <https://aws.amazon.com/es/what-is/rtt-in-networking/>
- Banerji, S., & Chowdhury, R. S. (2013). On IEEE 802.11: Wireless Lan Technology. *International Journal of Mobile Network Communications & Telematics*, 3(4), 45–64. <https://doi.org/10.5121/ijmnet.2013.3405>
- Bosk, M., Gajic, M., Schwarzmann, S., Lange, S., & Zinner, T. (2021). HTBQueue: A Hierarchical Token Bucket Implementation for the OMNeT++/INET Framework. *ArXiv*. <https://arxiv.org/pdf/2109.12879>
- Buñay, P., Pastor, D., Paguay, P., & Moreno, S. (2019). *novasinergia.unach.edu.ec*. *NOVASINERGIA*, 2. <https://doi.org/https://doi.org/10.37135/unach.ns.001.03.04>
- Chen, J., Liu, B., Zhou, H., Yu, Q., Gui, L., & Shen, X. S. (2017). QoS-Driven Efficient Client Association in High-Density Software-Defined WLAN. *IEEE Transactions on Vehicular Technology*, 66(8), 7372–7383. <https://doi.org/10.1109/TVT.2017.2668066>
- Cisco Systems. (2005). *DiffServ: The Scalable End-to-End QoS Model*. [https://www.cisco.com/en/US/technologies/tk543/tk766/technologies\\_white\\_paper09186a00800a3e2f.html](https://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.html)
- Cisco Systems. (2021). VNI Complete Forecast Highlights Global Internet Users: % of Population Devices and Connections per Capita Average Speeds Average Traffic per Capita per Month Global-Consumer Highlights. *VNI Complete Forecast Highlights Global Internet Users*.
- Codex, A. (2024, January 29). *reintech.io*. Configuring a Secure FTP Server with VSFTPD on Debian 12. <https://reintech.io/blog/configure-secure-ftp-server-vsftpd-debian-12>
- Cranley, N., & Davis, M. (2005). Performance Analysis of Network-level QoS with Encoding Configurations for Unicast Video Streaming over IEEE 802.11 WLAN Networks. *IEEE*. <https://doi.org/doi:10.1109/wirles.2005.1549461>

- Cronnor, C. (2023). *www.sdxcentralapirestful.com*. ¿Qué Son Las API de SDN Northbound (y Las API de REST de SDN)?  
<https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/north-bound-interfaces-api/>
- Dezfouli, B., Esmaeelzadeh, V., Sheth, J., & Radi, M. (2018). *A Review of Software-Defined WLANs: Architectures and Central Control Mechanisms*.  
<http://arxiv.org/abs/1809.00121>
- Dong, Z., Ming, Z., & Ming, X. (2014). Supporting “One Big AP” illusion in Enterprise WLAN: an SDN-based Solution. *IEEE*.
- Erickson, J. (2024, April 4). *www.oracle.com*. What Is JSON?  
<https://www.oracle.com/database/what-is-json/>
- F5. (2024). *www.f5.com*. GLOSSARY: CYBERSECURITY TERMS & DEFINITIONS What Is Network Availability? <https://www.f5.com/glossary/network-availability#:~:text=to%20user%20requests,-,What%20Is%20Network%20Availability%3F,performance%20demands%20placed%20on%20it.>
- Fortinet. (2022). *www.fortinet.com*. What Is Quality of Service (QOS) in Networking?  
<https://www.fortinet.com/resources/cyberglossary/qos-quality-of-service>
- Fortinet. (2024). *www.fortinet-tcp.com*. What Is Transmission Control Protocol TCP/IP?  
<https://www.fortinet.com/resources/cyberglossary/tcp-ip>
- Fouaz, B. (2021, November 30). *ubuntu.com*. Data Centre Networking: What Is OVS?  
<https://ubuntu.com/blog/data-centre-networking-what-is-ovs>
- García, T. (2007). *Análisis de los Modelos DiffServ e IntServ para brindar QoS en Internet*. [Universidad Tecnológica de la Mixteca]. [http://jupiter.utm.mx/~tesis\\_dig/10141.pdf](http://jupiter.utm.mx/~tesis_dig/10141.pdf)
- Geeks for Geeks. (2020). *www.geeksforgeeks.org*. What Is OSI Model? – Layers of OSI Model. <https://www.geeksforgeeks.org/open-systems-interconnection-model-osi/>
- Geeks for Geeks. (2023, June 2). *www.geeksforgeeks-buffer.org*. Performance of a Network.
- Gonzalez, J. (2020). *Desarrollo y análisis de mecanismos de calidad de servicio sobre redes definidas por software*. <https://github.com/joagonzalez/sdn-qos>
- Goodwin, M. (2023, November 27). *www.ibm.com*. What Is an SLA?  
<https://www.ibm.com/topics/service-level-agreement>
- Graham, A. (2019, April). *developer.ibm.com*. What Is CURL and How Does It Relate to APIs? <https://developer.ibm.com/articles/what-is-curl-command/>
- Hajlaoui, N., & Jabri, I. (2012). WMNC 2012 : proceedings of 2012 5th Joint IFIP Wireless and Mobile Networking Conference : September 19-20, 2012, Bratislava, Slovakia. *On the Performance of IEEE 802.11n Protocol*. <https://sci-hub.hkvisa.net/10.1109/WMNC.2012.6416145>



- Huawei. (2023). *support.huawei.com*. Quality of Service (QoS).  
<https://support.huawei.com/enterprise/en/doc/EDOC1100086518>
- Huawei Forums. (2021, December 19). *forum.huawei.com*. SSID, ESSID and BSSID.  
<https://forum.huawei.com/enterprise/en/ssid-essid-and-bssid/thread/667251801893781504-667213855346012160>
- IBM. (2023). *www.ibmapi.com*. ¿Qué Es Una API (Interfaz de Programación de Aplicaciones)? <https://www.ibm.com/topics/api>
- IEEE. (2009). *IEEE 802.11n Standard for Information Technology Telecommunications and information exchange between systems LANs and MANs Specific requirements. Part 11: WLAN MAC and PHY Specifications. Amendment 5: Enhancements for Higher Throughput*. <https://doi.org/10.1109/IEEESTD.2009.5307322>
- IEEE SA. (2023, May 16). *IEEE Standard Association*. Retrieved from The Evolution of Wi-Fi Technology and Standards. <https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/#:~:text=Since%20its%20introduction%20in%201997,802.11%20series%20published%20in%202021>
- Issabel. (2024). *www.issabel.com*. Issabel Es Impulsado Por Asterisk.  
<https://www.issabel.com/en/about-issabel/>
- ITU Academy. (2019). *academycourses.itu.int*. Introduction to Service Quality Regulation.  
<https://academycourses.itu.int/course/view.php?id=1595>
- ITU-T E.800. (2008). *UIT-T Rec. E.800 (09/2008) Definiciones de términos relativos a la calidad de servicio*.
- ITU-T Y.1540. (2019). *Recomendación UIT-T Y.1540 (12/2012) – Servicio de comunicación de datos con protocolo Internet – Parámetros de calidad de funcionamiento relativos a la disponibilidad y la transferencia de paquetes del protocolo Internet*.  
<http://handle.itu.int/11.1>
- ITU-T Y.1541. (2011). *Network performance objectives for IP-based services*.
- Jellyfin. (2018). *jellyfin.org*. About Jellyfin. <https://jellyfin.org/docs/general/about/>
- Karakus, M., & Durrezi, A. (2017). Quality of Service (QoS) in Software Defined Networking (SDN): A survey. In *Journal of Network and Computer Applications*. Academic Press. <https://doi.org/10.1016/j.jnca.2016.12.019>
- Kaspersky. (2024). *www.kaspersky.com*. What Is an SSL Certificate – Definition and Explanation. <https://www.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate>
- Keshari, S. K., Kansal, V., & Kumar, S. (2021). A Systematic Review of Quality of Services (QoS) in Software Defined Networking (SDN). *Wireless Personal Communications*, 116(3), 2593–2614. <https://doi.org/10.1007/s11277-020-07812-2>
- Kurose, J. F., & Ross, K. W. (2017). *Computer networking : A top-down approach* (Pearson, Ed.; 7th ed.).

- Laassiri, F., Idboufker, N., & Moughit, M. (2017). Evaluation of the QoS parameters in different SDN architecture using Omnet 4.6 ++. *IEEE*.
- Loshin, P., Linthicum, D., & Giza, M. (2024). *www.techtargetxml.com*. DEFINITION XML (Extensible Markup Language). <https://www.techtarget.com/whatis/definition/XML-Extensible-Markup-Language>
- Macedo, D. F., Guedes, D., Vieira, L. F. M., Vieira, M. A. M., & Nogueira, M. (2015). Programmable networks-From software-defined radio to software-defined networking. *IEEE Communications Surveys and Tutorials*, 17(2), 1102–1125. <https://doi.org/10.1109/COMST.2015.2402617>
- Malik, A., Qadir, J., Ahmad, B., Alvin Yau, K. L., & Ullah, U. (2015). QoS in IEEE 802.11-based wireless networks: A contemporary review. In *Journal of Network and Computer Applications* (Vol. 55, pp. 24–46). Academic Press. <https://doi.org/10.1016/j.jnca.2015.04.016>
- matplotlib-org. (2024, May 16). *matplotlib.org*. Matplotlib: Visualización Con Python. <https://matplotlib.org/>
- Moisa, J. (2021). *Tutors Academy*. QoS a Nivel de Capa 3. <https://www.youtube.com/watch?v=I6-SRs3j64U>
- Moncayo, D. (2023). *Implementación de un prototipo de Red Wireless Definida por Software Basado en Software Libre Aplicado en Ambientes de Laboratorios para la Carrera de Telecomunicaciones*.
- Murcia, O., & Beltrán, J. (2021). *Medición y comparación de parámetros de calidad de servicio en SDN*. Universidad Distrital Francisco José de Caldas.
- Mushtaq, A., & Singh, M. (2017). QOS Parameter Comparison of DiffServ-Aware MPLS Network Using IPv4 and IPv6. *Proceeding International Conference on Recent Innovations Is Signal Processing and Embedded Systems (RISE-2017) 27-29 October, 2017*.
- Nabar, R. (2014). MIMO in WiFi Systems. *Smart Antenna Workshop*. [https://web.stanford.edu/~apaulraj/workshop70/pdf/MIMO\\_WiFi\\_Nabar.pdf](https://web.stanford.edu/~apaulraj/workshop70/pdf/MIMO_WiFi_Nabar.pdf)
- Network Academy. (2023). *www.networkacademy.io*. IPv4 vs IPv6 - Understanding the Differences. <https://www.networkacademy.io/ccna/ipv6/ipv4-vs-ipv6>
- OffSec. (2024). *www.kali.org*. Documentación de La Herramienta: Hping3. <https://www.kali.org/tools/hping3/>
- ONF. (2014, June). *sdnarchitectureopennetworking.org*. SDN Architecture. [https://opennetworking.org/wp-content/uploads/2013/02/TR\\_SDN\\_ARCH\\_1.0\\_06062014.pdf](https://opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf)
- ONF. (2024). *ONF Open Source and Standards Liaisons & Collaborations*. <https://opennetworking.org/mission/>
- Powertec Wireless Technology. (2023). *Sistema MIMO de 4x4*. What Is 4x4 MIMO? <https://rspectrum.com.au/resources/wireless-transmission/mimo/4x4-mimo>

- Priya, S., & Singh, G. (2016). Comparison of Wi-Fi IEEE 802.11 Standards Relating to Media Access Control Protocols. In *Article in International Journal of Computer Science and Information Security*. <https://sites.google.com/site/ijcsis/>
- Python Documentation. (2024). *docs.python.org*. La Biblioteca Estándar de Python. <https://docs.python.org/es/3/library/index.html>
- Riggio, R., Rasheed, T., & Marina, M. K. (2014). Programming software-defined wireless networks. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 413–415. <https://doi.org/10.1145/2639108.2642897>
- Ryu. (2014). *ryu-github.com*. [https://osrg.github.io/ryu-book/en/html/rest\\_qos.html#id3](https://osrg.github.io/ryu-book/en/html/rest_qos.html#id3)
- Ryu Documentation. (2014). *ryu.readthedocs.io*. Getting Started. [https://ryu.readthedocs.io/en/latest/getting\\_started.html](https://ryu.readthedocs.io/en/latest/getting_started.html)
- Schmitt, J., & Wolf, L. (1997). Industrial Process and System Communications (KOM) Quality of Service-An Overview Quality of Service-An Overview. *Quality of Service - An Overview*. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=abf1fc3769c9f3c5b00e54c9accd1b35534536ed>
- SDxCentral. (2024). *www.sdxcentral.com*. ¿Qué Son Las API de SDN Southbound? <https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/southbound-interface-api/>
- Seitz, N. (2003). ITU-T QoS Standards for IP-Based Networks. *IEEE*. <https://doi.org/doi:10.1109/mcom.2003.1204752>
- Sheldon, R. (2024). *www.techtargetxslt.com*. XSL (Extensible Stylesheet Language). <https://www.techtarget.com/whatis/definition/XSL-Extensible-Stylesheet-Language>
- Shin, M.-K., Nam, K.-H., & Kim, H.-J. (2012). ICTC 2012 : 2012 International Conference on ICT Convergence : “Global Open Innovation Summit for Smart ICT Convergence,” October 15-17, 2012, Ramada Plaza Jeju Hotel, Jeju Island, Korea. *Software-Defined Networking (SDN): A Reference Architecture and Open APIs*. <https://doi.org/https://doi.org/10.1109/ICTC.2012.6386859>
- Soto, J. A. (2020, July 26). *www.geeknetic.es*. ¿Qué Es El Kernel y Para Qué Sirve? <https://www.geeknetic.es/Kernel/que-es-y-para-que-sirve>
- Stallings, W. (2014). *Data and Computer Communications: Vol. 10th Edition*. Pearson Education. <http://www.pearsonhighered.com/stallings/>
- Steed, A., & Fradinho, M. (2009). *Networked Graphics Building Networked Games and Virtual Environments*. Elsevier Inc. All rights reserved. <https://doi.org/https://doi.org/10.1016/C2009-0-19052-9>
- UNITED NATIONS DEVELOPMENT PROGRAMME. (2023). *Sustainable Development - Sustainable goals by 2030*. Sustainable Development - Sustainable Goals by 2030. <https://www.undp.org/sustainable-development-goals/no->

poverty?gclid=CjwKCAjwsvujBhAXEiwA\_UXnAG7lpSvHjgLsFEETL1I7lq01JKOA2kS4w0CiOWobKt1vwOMYk4oeOBoCM0UQAvD\_BwE

- Vasco, A. (2010). *DIMENSIONAMIENTO DE UNA CENTRAL TELEFÓNICA IP UTILIZANDO ESTÁNDARES ABIERTOS Y SOFTWARE LIBRE PARA LA EMPRESA CONECTIVIDAD GLOBAL*. <https://bibdigital.epn.edu.ec/bitstream/15000/2497/1/CD-3199.pdf>
- Visoottiviseth, V., Piroonsith, T., & Siwamogsatham, S. (2009). An Empirical Study on Achievable Throughputs of IEEE 802.11n Devices. *An Empirical Study on Achievable Throughputs of IEEE 802.11n Devices*. <https://doi.org/https://doi.org/https://eudl.eu/doi/10.1109/wiopt.2009.5291578>
- W3schools. (2024). *www.w3schools.com*. Pandas Introduction. [https://www.w3schools.com/python/pandas/pandas\\_intro.asp](https://www.w3schools.com/python/pandas/pandas_intro.asp)
- Zola, A. (2024). *www.techtargget.com*. What Is a Ping? <https://www.techtargget.com/searchnetworking/definition/ping>

## 8. ANEXOS

### 8.1. Anexo A: Instalación y configuración de servicios multimedia

Este apartado presenta la instalación de servicios multimedia definidos para la red SDWN. Los servicios incluyen Video bajo demanda (VOD), Transferencia de archivos (FTP) y Voz sobre IP (VoIP), mismos que se implementan en máquinas virtuales alojadas en el hipervisor VirtualBox. Los servicios VOD y FTP se instalan en una máquina virtual con el sistema operativo Debian 12, mientras que VoIP se ejecuta en una máquina con CentOS7. La **Tabla 44** muestra las características operativas de las máquinas mencionadas, y en las siguientes secciones se describe el proceso de instalación de cada servicio.

**Tabla 44**  
*Características de máquinas alojadas en VirtualBox.*

Característica	Máquinas Virtuales	
	Debian 12	CentOS 7
<b>Memoria RAM</b>	2 GB	2 GB
<b>Almacenamiento</b>	50 GB	50 GB
<b>Arquitectura</b>	64 bits	64 bits
<b>Procesador</b>	1 núcleo	1 núcleo
<b>Servicios alojados</b>	VOD FTP	VoIP

#### 8.1.1. Servicio VOD con Jellyfin

Jellyfin es un software de código abierto que permite la administración y reproducción de contenidos de video (Jellyfin, 2018). Una de sus principales características es su capacidad para adaptar la calidad de reproducción del video, funcionalidad que puede ser regulada tanto manual como de forma automática. Esta característica es particularmente útil en este proyecto, ya que permitirá conocer el comportamiento de la red en términos de QoS sobre este servicio. A continuación, se describen los pasos para la instalación de Jellyfin.

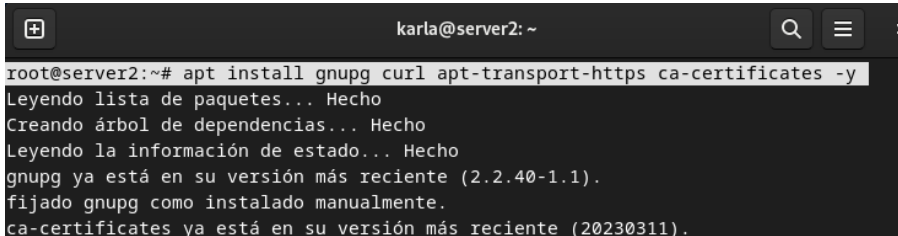
⇒ **Paso 1:** Instalación de prerequisites.

Los prerequisites son paquetes necesarios para la correcta instalación y operación de Jellyfin. En este caso se requieren cuatro paquetes específicos, los cuales se detallan en la **Tabla 45**. Por otro lado, el proceso de instalación de estos paquetes se ilustra en la **Figura 138**.

**Tabla 45**  
*Prerequisites para la instalación de Jellyfin.*

Paquete	Descripción
gnupg	Permite encriptar y firmar los datos del repositorio oficial de Jellyfin.
curl	Útil para descargar y actualizar paquetes necesarios para la operación de Jellyfin.
apt-transport-https	Es una extensión del gestor de paquetes apt que habilita la descarga segura de paquetes desde repositorios HTTPS.
ca-certificates	Es un paquete que contiene certificados de autoridades de certificación de confianza, necesarios para verificar la autenticidad de sitios web y servicios.

**Figura 138**  
*Instalación de dependencias para servicio Jellyfin.*



```

karla@server2: ~
root@server2:~# apt install gnupg curl apt-transport-https ca-certificates -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
gnupg ya está en su versión más reciente (2.2.40-1.1).
fijado gnupg como instalado manualmente.
ca-certificates ya está en su versión más reciente (20230311).

```

⇒ **Paso 2:** Configuración del Repositorio para Jellyfin

Para configurar este repositorio, primero se crea el directorio `/etc/apt/keyrings`. A continuación, se descarga y añade la clave GPG<sup>31</sup> de Jellyfin a dicho directorio. Estos pasos permiten la correcta configuración del repositorio creado y habilitan la descarga e instalación de Jellyfin utilizando el gestor de paquetes ‘apt’. El proceso descrito se presenta en la **Figura 139**.

<sup>31</sup> Clave GPG es un sistema permite cifrar y firmar digitalmente documentos y correos electrónicos para garantizar la seguridad y autenticidad de la información.

**Figura 139**

*Descarga de firma GPG del equipo de Jellyfin.*

```

root@server2:~# sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://repo.jellyfin.org/jellyfin_team.gpg.key | sudo gpg --dearmor
-o /etc/apt/keyrings/jellyfin.gpg
root@server2:~# ls /etc/apt/keyrings/
jellyfin.gpg
root@server2:~#

```

Una vez añadida la clave GPG, se debe agregar el repositorio de Jellyfin al archivo de fuentes del sistema operativo, ubicado en la ruta `‘/etc/apt/sources.list.d/jellyfin.list’`. Este archivo define las ubicaciones desde donde el sistema puede descargar los paquetes de software necesarios para la instalación y actualización de Jellyfin. En la **Figura 140** muestran las configuraciones específicas dentro del archivo mencionado. Además, se destaca que luego de haber configurado el archivo, es necesario actualizar el sistema operativo para asegurarse de que el gestor de paquetes `‘apt’` reconozca el nuevo repositorio y su contenido, tal proceso se realiza con el comando `apt-get update`.

**Figura 140**

*Edición de archivo `jellyfin.sources`.*

```

karla@server2: ~
GNU nano 7.2 /etc/apt/sources.list.d/jellyfin.sources *
Types: deb
URIs: https://repo.jellyfin.org/debian
Suites: bookworm
Components: main
Architectures: amd64
Signed-By: /etc/apt/keyrings/jellyfin.gpg

```

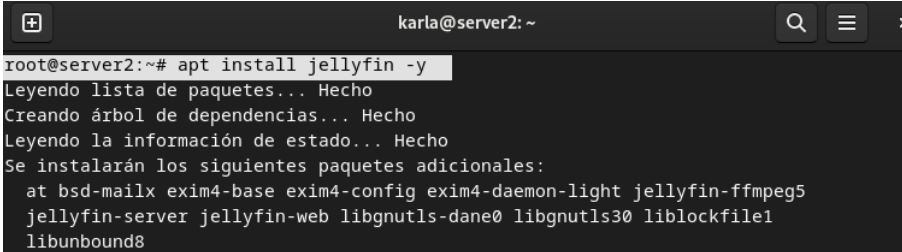
⇒ **Paso 3:** Instalación de Jellyfin

Con el repositorio de Jellyfin agregado, se procede a la instalación de este servicio. Para llevar a cabo este proceso se utiliza el gestor de paquetes `apt`, donde se emplea el comando `apt install jellyfin -y`. El parámetro `-y` se incluye para que la instalación se confirme automáticamente, evitando así la necesidad de intervención manual. La **Figura 141** la ejecución de este comando.

A continuación, se pone en marcha el servicio y se configura para que se inicie de forma automática con cada arranque del sistema. Para iniciar el servicio, se utiliza el comando ‘systemctl start jellyfin’, y para habilitarlo en el arranque del sistema, se emplea el comando ‘systemctl enable jellyfin’. Estos pasos se ilustran en la **Figura 142**.

**Figura 141**

*Instalación de Jellyfin.*

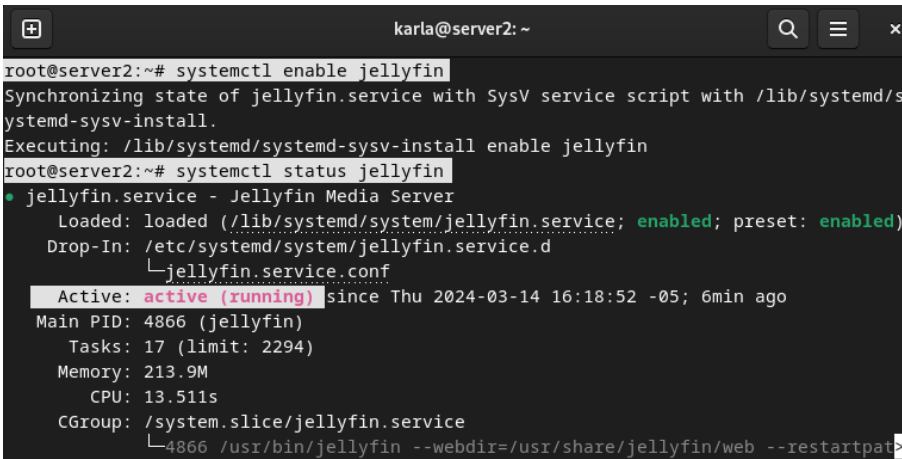


```

karla@server2: ~
root@server2:~# apt install jellyfin -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  at bsd-mailx exim4-base exim4-config exim4-daemon-light jellyfin-ffmpeg5
  jellyfin-server jellyfin-web libgnutls-dane0 libgnutls30 liblockfile1
  libunbound8
  
```

**Figura 142**

*Administración de servicio Jellyfin.*



```

karla@server2: ~
root@server2:~# systemctl enable jellyfin
Synchronizing state of jellyfin.service with SysV service script with /lib/systemd/s
ysystemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jellyfin
root@server2:~# systemctl status jellyfin
• jellyfin.service - Jellyfin Media Server
  Loaded: loaded (/lib/systemd/system/jellyfin.service; enabled; preset: enabled)
  Drop-In: /etc/systemd/system/jellyfin.service.d
           └─jellyfin.service.conf
  Active: active (running) since Thu 2024-03-14 16:18:52 -05; 6min ago
  Main PID: 4866 (jellyfin)
  Tasks: 17 (limit: 2294)
  Memory: 213.9M
  CPU: 13.511s
  CGroup: /system.slice/jellyfin.service
          └─4866 /usr/bin/jellyfin --webdir=/usr/share/jellyfin/web --restartpat>
  
```

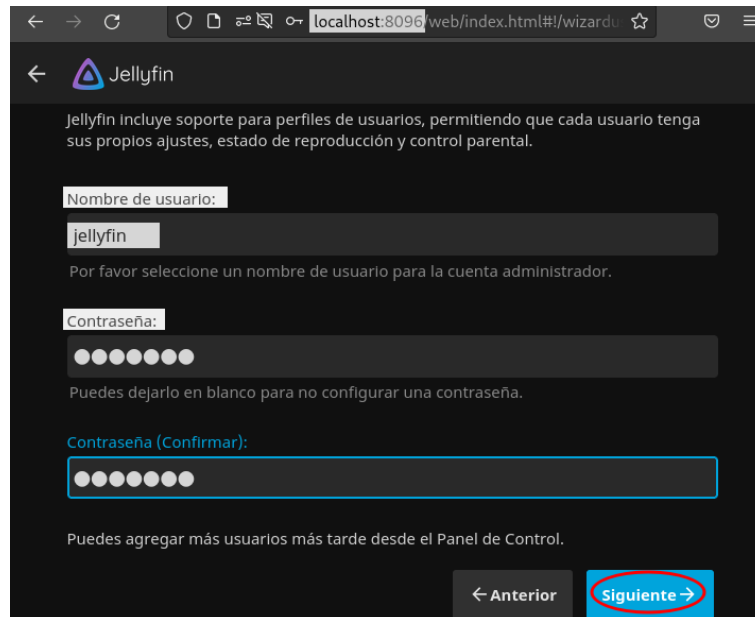
⇒ **Paso 4:** Configuraciones iniciales de Jellyfin desde la interfaz web

Jellyfin posee una interfaz web que opera en el puerto 8096. Para acceder a ella desde un navegador, se debe ubicar la dirección IP del servidor seguida del puerto mencionado anteriormente. Al ingresar por primera vez a esta interfaz, se requiere configurar el idioma y crear un usuario de acceso. En este caso, el idioma seleccionado es Español/Latioamérica y el proceso de creación del usuario se evidencia en la **Figura 143**.



**Figura 143**

Creación de usuario para acceder a interfaz web de Jellyfin.

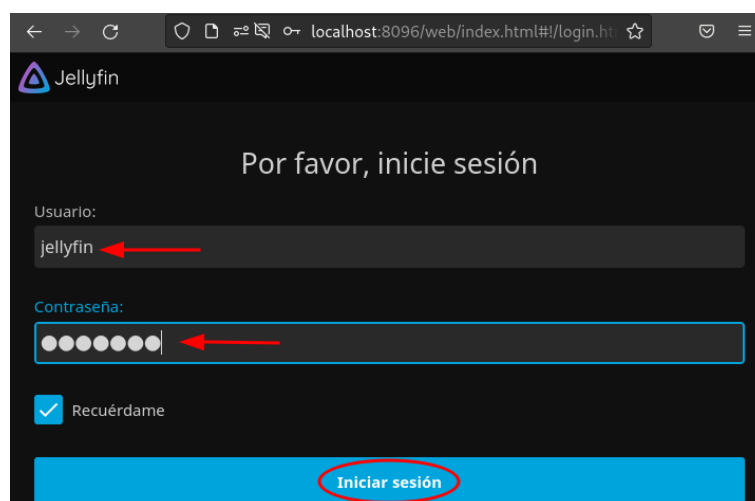


The screenshot shows a web browser window at localhost:8096/web/index.html#/wizard. The page title is 'Jellyfin'. The main heading is 'Jellyfin Incluye soporte para perfiles de usuarios, permitiendo que cada usuario tenga sus propios ajustes, estado de reproducción y control parental.' Below this, there are three input fields: 'Nombre de usuario:' with the value 'jellyfin', 'Contraseña:' with a masked password, and 'Contraseña (Confirmar):' with a masked password. A message below the password fields says 'Puedes dejarlo en blanco para no configurar una contraseña.' At the bottom, there are two buttons: '← Anterior' and 'Siguiente →', with the latter circled in red.

Para verificar las configuraciones anteriores, es necesario volver a ingresar a la interfaz web de Jellyfin. Al ingresar, se surgirá una ventana de "Login" (ver **Figura 144**), misma que facilitará el inicio de sesión utilizando las credenciales del usuario previamente creadas. Este proceso garantiza que las configuraciones realizadas anteriormente están operativas.

**Figura 144**

Inicio de sesión en GUI web de Jellyfin.



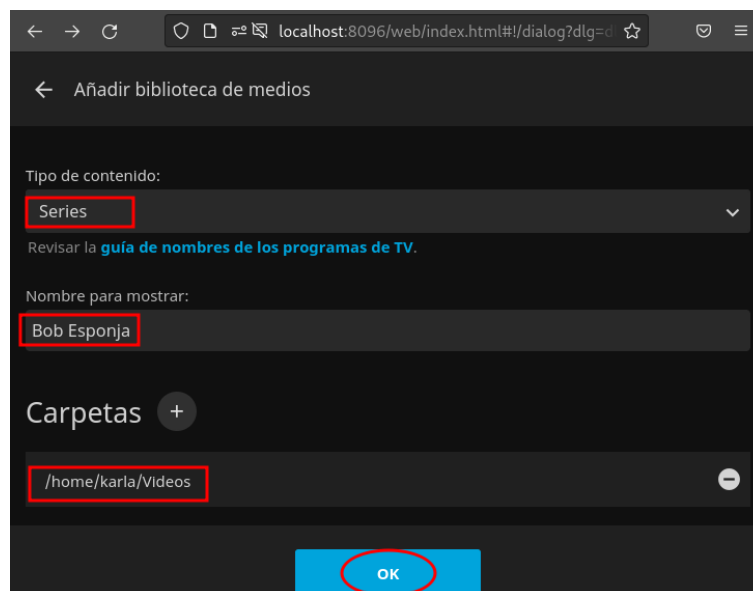
The screenshot shows a web browser window at localhost:8096/web/index.html#/login. The page title is 'Jellyfin'. The main heading is 'Por favor, inicie sesión'. Below this, there are two input fields: 'Usuario:' with the value 'jellyfin' and 'Contraseña:' with a masked password. A checkbox labeled 'Recuérdame' is checked. At the bottom, there is a blue button labeled 'Iniciar sesión', which is circled in red.

⇒ **Paso 5:** Adición de contenido multimedia a servidor

El último paso consiste en la adición de videos al servidor VOD. Para llevar a cabo esta tarea, los videos deben ubicarse en una ruta específica dentro del directorio home, asegurando así que sean accesibles por el servidor. Así, desde la interfaz de Jellyfin se debe ingresar en el menú ‘añadir biblioteca de medios’, en este apartado es necesario brindar información acerca del contenido multimedia, incluida la carpeta donde se encuentra almacenado dicho contenido (ver **Figura 145**).

**Figura 145**

*Adición de contenido multimedia a servidor.*



Finalmente, se verifica que el contenido sea accesible para los usuarios. Esto se realiza ingresando remotamente al servidor a través de la interfaz web de Jellyfin. Así, la **Figura 146** comprueba que los videos añadidos son visibles y se reproducen sin inconvenientes.

**Figura 146**

*Visualización de contenido multimedia.*



### **8.1.2. Servicio de Transferencia de Archivos (FTP)**

Para la implementación del servicio de transferencia de archivos (FTP), se utiliza el daemon<sup>32</sup> VSFTPD (Very Secure FTP Daemon). Este daemon es un servidor FTP que se caracteriza por brindar alto rendimiento, estabilidad y uso eficientemente los recursos del sistema. Dichas características lo hacen adecuado para operar en servidores con limitaciones de hardware (Codex, 2024). Además, la instalación de este daemon no es compleja y se detalla en los siguientes cinco pasos.

#### **⇒ Paso 1: Instalación y administración de VSFTPD**

Para llevar a cabo la instalación de VSFTPD se emplea el gestor de paquetes apt, sin necesidad de configurar repositorios adicionales como se realizó en apartados anteriores. Así el comando de instalación es `apt install vsftpd` (ver **Figura 147**). Luego de la instalación, se debe iniciar el servicio y configurarlo para que se ejecute automáticamente con cada arranque del sistema, esto se ilustra en la **Figura 148**.

<sup>32</sup> Daemon es un término utilizado en sistemas operativos Linux para referirse a un programa que se ejecuta en segundo plano y realiza tareas específicas sin intervención directa del usuario.

**Figura 147**  
*Instalación de servicio vsftpd.*

```

karla@server2: ~
root@server2:~# sudo apt install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  vsftpd
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 199 no actualizados.
Se necesita descargar 142 kB de archivos.
Se utilizarán 351 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 vsftpd amd64 3.0.3-13+b2 [
142 kB]
Descargados 142 kB en 1s (215 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete vsftpd previamente no seleccionado.
(Leyendo la base de datos ... 156983 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../vsftpd_3.0.3-13+b2_amd64.deb ...
Desempaquetando vsftpd (3.0.3-13+b2) ...
Configurando vsftpd (3.0.3-13+b2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.serv
ice -> /lib/systemd/system/vsftpd.service.
Procesando disparadores para man-db (2.11.2-2) ...

```

**Figura 148**  
*Administración del servicio.*

```

root@server2:~# sudo systemctl is-enabled vsftpd
enabled
root@server2:~# sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; preset: enable>
   Active: active (running) since Sat 2024-03-23 16:35:05 -05; 10min ago
   Process: 4517 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited,
   Main PID: 4518 (vsftpd)
   Tasks: 1 (limit: 2294)
   Memory: 1.4M
   CPU: 12ms
   CGroup: /system.slice/vsftpd.service
           └─4518 /usr/sbin/vsftpd /etc/vsftpd.conf

mar 23 16:35:05 server2 systemd[1]: Starting vsftpd.service - vsftpd FTP server.
mar 23 16:35:05 server2 systemd[1]: Started vsftpd.service - vsftpd FTP server.
lines 1-13/13 (END)
root@server2:~#

```

⇒ **Paso 2:** Configuración de acceso al servidor FTP

Esta configuración permite establecer un marco de seguridad que controle el acceso al servidor. Este proceso se realiza en el archivo de configuración ubicado en la ruta `/etc/vsftpd.conf`, en este archivo se definen las directrices y parámetros necesarios para regular la autenticación de usuarios, permisos, y restricciones de seguridad (ver **Tabla 46**). Una vez aplicadas estas configuraciones, es necesario reiniciar el servicio para que el sistema las reconozca, esto se evidencia en la **Figura 149**.

**Tabla 46**  
Configuración de seguridad y acceso al servidor FTP.

Configuración	Función
anonymous_enable=NO	Deshabilita el acceso anónimo.
local_enable=YES	Habilita el acceso de usuarios locales.
write_enable=YES	Facilita la gestión de archivos por parte de los usuarios.
chroot_local_user=YES	Limita el acceso de usuarios locales a su directorio de inicio, impidiendo que puedan acceder a otras partes del sistema de archivos.
local_root=/home/\$USER/ftp	Define el directorio raíz de FTP para cada usuario.
userlist_enable=YES	Habilita el uso de una lista de usuarios especificada que tienen acceso al servidor.
userlist_file=/etc/vsftpd.userlist	Especifica la ruta del archivo que contiene la lista de usuarios permitidos.

**Figura 149**  
Aplicación de configuraciones realizadas.

```

root@server2:~# sudo systemctl restart vsftpd
root@server2:~# sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; preset: enable>
   Active: active (running) since Sat 2024-03-23 16:59:14 -05; 7s ago
   Process: 4720 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited,
   Main PID: 4721 (vsftpd)
   Tasks: 1 (limit: 2294)
   Memory: 1.0M
   CPU: 14ms
   CGroup: /system.slice/vsftpd.service
           └─4721 /usr/sbin/vsftpd /etc/vsftpd.conf

mar 23 16:59:14 server2 systemd[1]: Starting vsftpd.service - vsftpd FTP server.
mar 23 16:59:14 server2 systemd[1]: Started vsftpd.service - vsftpd FTP server.

```

⇒ **Paso 3:** Creación de usuario y directorio para acceso a FTP

En las configuraciones anteriores se detalló que el acceso al servidor se gestiona mediante el archivo `vsftpd.source.list`, así se procede a crear un usuario para añadirlo a lista de usuarios permitidos. Para la creación de usuario se emplea el comando `adduser` seguido del nombre o identificativo del usuario, posteriormente, se configura una contraseña y, si es necesario, se pueden especificar más datos del usuario, tal como se muestra en la **Figura 150**. Finalmente, se agrega este usuario al archivo `source.list`, esto se presenta en la **Figura 151**.

**Figura 150**  
Creación de usuario para acceso a FTP.

```

root@server2:~# sudo adduser tesis2024
Añadiendo el usuario `tesis2024' ...
Añadiendo el nuevo grupo `tesis2024' (1001) ...
Adding new user `tesis2024' (1001) with group `tesis2024 (1001)' ...
Creando el directorio personal `/home/tesis2024' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para tesis2024
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
Nombre completo []: Tesis QoS
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
Adding new user `tesis2024' to supplemental / extra groups `users' ...
Añadiendo al usuario `tesis2024' al grupo `users' ...
root@server2:~#

```

**Figura 151**  
Adición de usuario FTP al archivo userlist.

```

root@server2:~# echo "tesis2024" | sudo tee -a /etc/vsftpd.userlist
tesis2024
root@server2:~# cat /etc/vsftpd.userlist
tesis2024
root@server2:~# sudo systemctl restart vsftpd
root@server2:~# sudo systemctl status vsftpd
root@server2:~# sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-03-23 17:25:55 -05; 6s ago
     Process: 4946 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0)
    Main PID: 4947 (vsftpd)
      Tasks: 1 (limit: 2294)
     Memory: 1.1M
        CPU: 7ms
    CGroup: /system.slice/vsftpd.service
            └─4947 /usr/sbin/vsftpd /etc/vsftpd.conf

```

Lo siguiente es crear el directorio que será compartido en FTP. En este caso debe crearse específicamente en la ruta `/etc/home/nombre-usuario`, debido a que es el único acceso permitido para FTP, conforme a las configuraciones establecidas en el **Paso 2**. Además, los permisos del directorio y de los archivos que contenga, deben especificarse de la siguiente manera: (i) propietario: usuario y (ii) grupo: usuario. Lo antes descrito se muestra en la **Figura 152**.

**Figura 152**  
Creación de directorio compartido con FTP.

```

root@server2:~# sudo mkdir /home/tesis2024/ftp/files/
root@server2:~# sudo chown tesis2024:tesis2024 /home/tesis2024/ftp/files
root@server2:~# sudo ls -la /home/tesis2024/ftp
total 12
dr-xr-xr-x 3 nobody nogroup 4096 mar 23 17:20 .
root@server2:~# echo "archivo de prueba" | sudo tee /home/tesis2024/ftp/files/test.txt
archivo de prueba
root@server2:~#

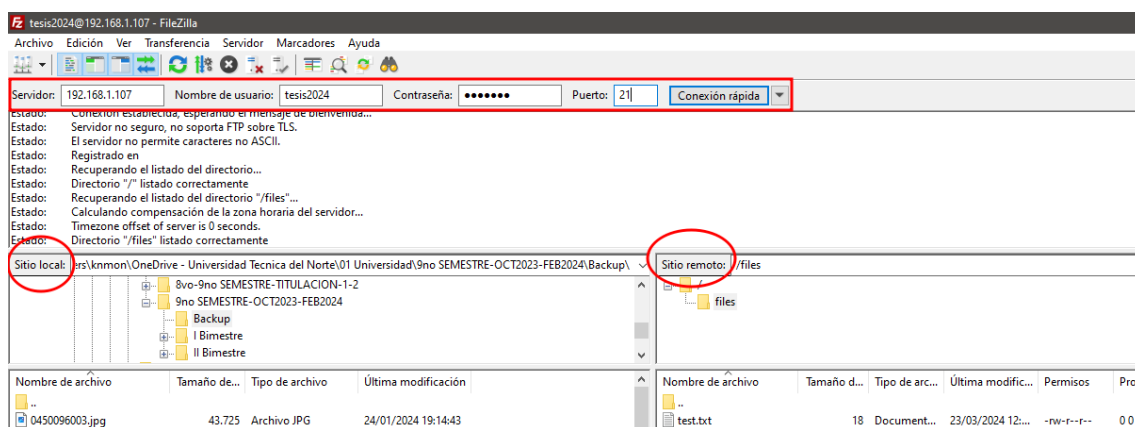
```

#### ⇒ Paso 4: Acceso a servidor FTP

Por último, para validar la configuración establecida en los pasos previos, se realiza el acceso al servidor FTP a través del cliente Filezilla. En la **Figura 153** se evidencia que el servidor se encuentra operativo y funcionando adecuadamente.

**Figura 153**

*Acceso a servidor FTP median agente FileZilla.*



#### 8.1.3. Servicio de Voz sobre IP (VoIP) con Issabel

Issabel es un PBX<sup>33</sup> de código abierto que proporciona configuración, administración e informes para un sistema de telefonía. Es compatible con el codec G.711, el cual es un estándar de compresión de voz empleado en VoIP. Las características de este codec es comunicación clara y de alta calidad en entornos empresariales y de pequeñas empresas. Para utilizar este software, la comunidad que desarrolló Issabel ofrece una máquina virtual que puede descargarse de forma gratuita (Issabel, 2024). En las siguientes secciones, se detalla el proceso de descarga y administración de este recurso.

<sup>33</sup> PBX (Private Branch Exchange), es un sistema de enrutamiento telefónico privado que conecta varias extensiones de una oficina entre sí y a un pequeño número de líneas externas.

⇒ **Paso 1:** Descarga de máquina virtual Issabel

La descarga se realiza desde la página oficial de Issabel. La máquina virtual se encuentra disponible en la sección de descargas, donde es posible acceder al archivo compatible con el entorno de virtualización que emplee. La **Figura 154** evidencia este proceso.

**Figura 154**

*Descarga de máquina virtual Issabel.*

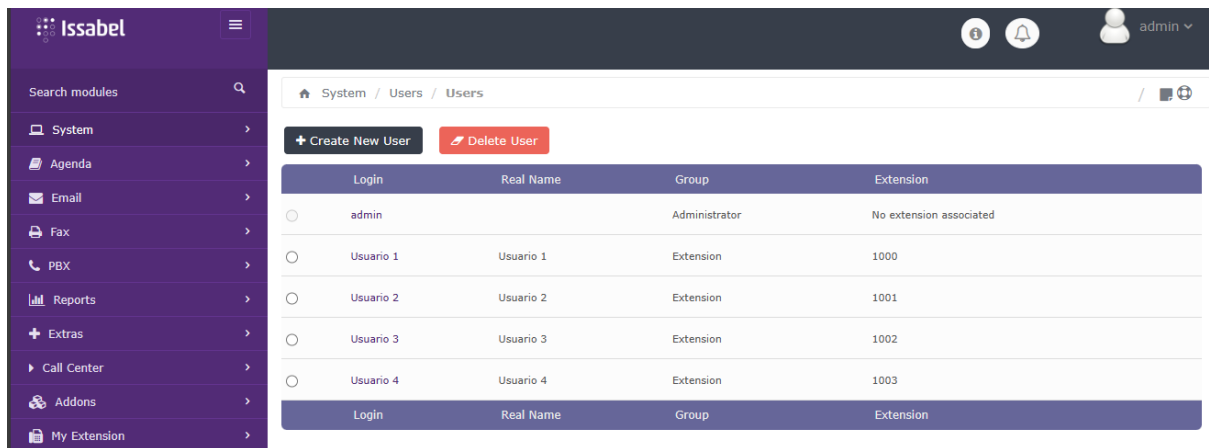


⇒ **Paso 2:** Configuración de usuarios y extensiones

La configuración de Issabel consiste en la creación de extensiones y usuarios para permitir el acceso al sistema de telefonía. Este proceso se lleva a cabo a través de la interfaz web, a la cual se accede mediante la dirección IP del servidor Issabel, luego desde el menú principal se puede acceder a la sección "PBX", donde se realizan estas configuraciones. En la **Figura 155** se presenta las configuraciones realizadas.



**Figura 155**  
*Usuarios y extensiones creadas.*



Login	Real Name	Group	Extension
admin		Administrator	No extension associated
Usuario 1	Usuario 1	Extension	1000
Usuario 2	Usuario 2	Extension	1001
Usuario 3	Usuario 3	Extension	1002
Usuario 4	Usuario 4	Extension	1003

⇒ **Paso 3: Acceso al servicio de VoIP**

Para finalizar, se procede a verificar el funcionamiento del servidor VoIP. Para llevar a cabo esta tarea, se hace uso del softphone Zoiper, el cual permite conectarse al servidor utilizando los usuarios y extensiones configuradas. Así, en la **Figura 156**, se muestra el correcto funcionamiento del servidor, evidenciando tanto el ingreso como la gestión de llamadas.

**Figura 156**  
*Funcionamiento de servicio VoIP.*

