



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

“ANÁLISIS DEL DESEMPEÑO DE MPLS VPN L2 y L3”

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN
INGENIERÍA DE REDES DE COMUNICACIONES

JORGE EDUARDO FLORES BALDÉS

PROFESOR GUÍA:

CLAUDIO IGNACIO ESTÉVEZ MONTERO

MIEMBROS DE LA COMISIÓN:

JORGE IGNACIO SANDOVAL ARENAS

ÁLVARO ARMANDO SILVA MADRID

SANTIAGO DE CHILE

2018

RESUMEN DE LA TESIS PARA OPTAR AL
TÍTULO DE MAGÍSTER EN INGENIERÍA DE REDES DE COMUNICACIONES
POR: JORGE EDUARDO FLORES BALDÉS
FECHA: 2018
PROF. GUÍA: SR. CLAUDIO ESTÉVEZ MONTERO

ANÁLISIS DEL DESEMPEÑO DE MPLS VPN L2 y L3

La conmutación de etiquetas multiprotocolo (MPLS por sus siglas en inglés, *Multiprotocol Label Switching*) surge como un mecanismo de convergencia para los protocolos que operan sobre los niveles 2 y 3 del modelo OSI. Su capacidad para proveer y administrar diversos servicios con garantías de calidad de servicio y disponibilidad sobre una infraestructura común, ha hecho que MPLS sea un estándar en las redes de transporte de los proveedores de servicios.

La interconexión de *data centers* y en general de redes LAN y MAN corporativas, se realiza a través de servicios MPLS VPN considerando solamente la topología de la red. En este contexto, resulta útil proporcionar información adicional para seleccionar modelos VPN en función del tipo de tráfico que se desea transportar.

En este trabajo se diseñan e implementan escenarios experimentales para proporcionar métricas que permiten ese contraste; el resumen de cada capítulo se detalla a continuación.

En el primer capítulo se describen tecnologías, métricas de desempeño, herramientas de modelación y herramientas estadísticas.

En el segundo capítulo se describen los procesos de diseño, implementación y simulación de los escenarios experimentales. Los escenarios garantizan que el desempeño de los servicios MPLS VPN se ponga a prueba bajo las mismas condiciones. Esas condiciones comprenden nodos de borde y políticas de QoS comunes para los servicios que se contrastan. Además, los escenarios consideran la capacidad de los nodos emulados por Dynamips como restricción y el tráfico que atraviesa una red operativa como condición inicial. Este tráfico se modela con redes neuronales artificiales y para poder generarlo con IPERF, se utiliza BoxCox y Bootstrapping sobre el modelo para obtener estadísticos representativos. Los procesos de implementación y simulación se realizan sobre GNS3; este último comprende la ejecución simultánea y recurrente de IP SLA, kron, IPERF, Wireshark, NTP y TFTP.

En el tercer capítulo se presenta el resultado de los test estadísticos aplicados sobre las métricas de estudio. Además, se utilizan herramientas de simulación para estimar los intervalos de confianza de la media y obtener una representación gráfica del desempeño de los servicios MPLS VPN.

En el cuarto capítulo se exponen las conclusiones de este trabajo, estas analizan los resultados de los test estadísticos asociados a los objetivos e hipótesis planteadas. Para finalizar se exponen algunas apreciaciones sobre trabajos futuros.

A mi madre, mujer sabia, de principios y valores, quien con infinito amor y esfuerzo ha guiado acertadamente cada paso en mi vida y ha dejado un legado más grande que cualquier bien material. Mamita Rosa, el reconocimiento es para usted.

Agradecimientos

A mis segundos padres, Sr. Rolando Heredia y Sra. Fanny Real, por todo el apoyo brindado.

A Mishi y Sonita, por el amor y cuidados entregados a mi madre durante mi estadía en Chile. Estoy en deuda con ustedes.

A mis compañeros de magíster, en especial al “clan” 47H, con quienes compartimos gratos momentos y se cultivó una gran amistad.

Al claustro académico, en especial a los profesores Alberto Castro, Claudio Estévez, Sergio Miranda, Jorge Sandoval y Álvaro Silva, por haber compartido desinteresadamente sus conocimientos y experiencia conmigo.

Tabla de contenido

Introducción	1
1. Motivación	1
2. Hipótesis.....	1
3. Objetivos	1
3.1. Objetivo principal.....	1
3.2. Objetivos específicos.....	1
4. Metodología de la investigación	1
Capítulo 1	3
Marco Teórico	3
1.1. MPLS	3
1.1.1. Descripción y funcionamiento.....	3
1.1.2. Plano de control.....	4
1.1.3. Cabecera MPLS.....	4
1.1.4. MPLS VPN	5
1.1.5. Calidad de servicio – QoS	7
1.2. Métricas de desempeño en redes IP	7
1.2.1. Métricas IETF	8
1.2.2. Métricas ITU-T	9
1.3. Técnicas de monitoreo activo y pasivo	9
1.4. Redes Neuronales Artificiales Feed-forward	9
1.5. Test de Anderson-Darling	10
1.6. Bootstrapping	11
1.7. Transformada de Box-Cox	11
1.8. Prueba de U Mann-Whitney.....	12
1.9. R Statistics.....	12
Capítulo 2	13
Metodología	13
2.1. Diseño	13
2.1.1. Topología de los escenarios	13
2.1.2. Métricas de desempeño	14
2.1.3. Condiciones Iniciales	14
2.1.4. Herramientas	17

2.1.5.	Restricciones	18
2.2.	Implementación.....	19
2.2.1.	Configuración MPLS	19
2.2.2.	Unicast.....	19
2.2.3.	Multicast.....	20
2.2.4.	Configuración NTP	21
2.2.5.	Inyección del tráfico de <i>backbone</i>	22
2.2.6.	Configuración de políticas de QoS.....	22
2.2.7.	Configuración WinAgents TFTP	23
2.3.	Simulación.....	23
Capítulo 3	25
Análisis de datos	25
3.1.	Test de Normalidad	25
3.2.	Prueba U de Mann-Whitney.....	26
3.2.1.	Métricas unicast.....	26
3.2.2.	Métricas multicast	27
3.3.	Intervalos de confianza (I.C.) de la media.....	29
3.3.1.	Métricas unicast.....	29
3.3.2.	Métricas multicast	32
Capítulo 4	35
Conclusiones	35
4.1.	Objetivos	35
4.2.	Metodología	35
4.3.	Hipótesis.....	35
4.4.	Trabajos futuros.....	36
Bibliografía	39
Anexos	40
A.	Códigos R.....	40
1.	Modelación del tráfico con redes neuronales artificiales	40
2.	Transformación con Box-Cox y estimación de la media poblacional.....	45
3.	Test no paramétricos y estimación de los I.C. de la media	46

Introducción

1. Motivación

MPLS es un mecanismo de transporte que utiliza etiquetas para tomar decisiones sobre el reenvío de datos. No tiene dependencia con tecnologías subyacentes, por lo tanto, permite desplegar servicios *any-to-any* y administrar políticas de calidad de servicio (QoS) de forma convergente. Las marcas de QoS impuestas por el cliente tienen su equivalente en el dominio MPLS, por lo tanto, el proveedor puede confiar en el marcaje o re-marcar los paquetes para garantizar QoS end-to-end. Además, MPLS soporta redes privadas virtuales (VPN), mecanismos de ingeniería de tráfico y recuperación ante fallos (FRR).

En el contexto de las VPN, MPLS soporta varios modelos de conexión y pueden clasificarse de acuerdo al nivel de participación del proveedor de servicios en el enrutamiento del cliente: nivel 2 (VPN L2) y nivel 3 (VPN L3).

Los documentos de referencia de los fabricantes de equipos sugieren el despliegue de los modelos VPN considerando únicamente la topología de la red y omiten un contraste de métricas que resultan importantes cuando se proporcionan servicios sensibles a la latencia y pérdida de paquetes. Aunque esto puede gestionarse con políticas de QoS, el despliegue de modelos adecuados puede mejorar el desempeño de los servicios que se transportan sobre MPLS VPN. Esta es la principal motivación para realizar este trabajo de grado.

Es necesario recalcar que el contraste de métricas en sistemas que requieren criterios de selección no aplica solamente a servicios MPLS VPN, por lo tanto, puede presentarse como una pauta para trabajos futuros como SD-WAN vs WAN, etc.

2. Hipótesis

El desempeño de los servicios MPLS VPN L2 en términos de delay, packet loss, jitter, MOS e ICPIF es mejor comparado con los servicios MPLS VPN L3.

3. Objetivos

3.1. Objetivo principal

Cuantificar y comparar el desempeño de los servicios MPLS VPN L2 y L3.

3.2. Objetivos específicos

- 3.2.1. Diseñar los escenarios de simulación y establecer las métricas de estudio.
- 3.2.2. Modelar el comportamiento del tráfico de control y usuario de un ISP.
- 3.2.3. Implementar los escenarios de simulación.
- 3.2.4. Incorporar el tráfico modelado en los escenarios y ejecutar las simulaciones.
- 3.2.5. Analizar las métricas obtenidas post-simulación utilizando test estadísticos.

4. Metodología de la investigación

Para lograr los objetivos planteados se utiliza una serie de procesos enmarcados dentro del método experimental y la investigación exploratoria. Los procesos incluyen:

Estudios exploratorios: Se revisa bibliografía específica y en conjunto con el criterio de expertos se diseñan los experimentos; estos incluyen topología de los escenarios, condiciones iniciales y métricas de estudio.

Modelación: Mediante el uso de redes neuronales artificiales se modela el comportamiento del tráfico de control y usuario de un ISP.

Implementación y simulación: Se implementan los escenarios en el simulador GNS3, se inyecta el tráfico modelado en los escenarios y finalmente se ejecutan las simulaciones.

Análisis de datos: Post-simulación se analizan las métricas obtenidas utilizando herramientas estadísticas.

Capítulo 1

Marco Teórico

En este capítulo se describen los recursos utilizados para el desarrollo de este trabajo. Estos recursos incluyen: tecnologías, métricas de desempeño, herramientas de modelación y herramientas estadísticas.

1.1. MPLS [1]

1.1.1. Descripción y funcionamiento

En una red MPLS, el envío de paquetes se basa en etiquetas. Las etiquetas pueden estar basadas en prefijos o en otro tipo de parámetros y son generadas por cada uno de los nodos que forman la red MPLS. El objeto de las etiquetas es formar una trayectoria entre dos o más nodos de borde. Esta trayectoria se denomina LSP y es unidireccional.

Según sus funciones, una red MPLS está compuesta por dos tipos de nodos: enrutadores de frontera de etiquetado (LER por sus siglas en inglés, *Label Edge Routers*) y enrutadores de conmutación de etiquetas (LSR por sus siglas en inglés, *Label Switching Routers*).

LER / PE: Están ubicados en el borde del dominio MPLS y son los únicos que analizan sus tablas de enrutamiento para determinar si una red es o no alcanzable (*Routing lookup*). Sus funciones en el dominio MPLS son adicionar y retirar etiquetas.

LSR / P: Son los encargados de realizar la conmutación de etiquetas. Grosso modo, un LSR recibe un paquete etiquetado, intercambia la etiqueta y envía el paquete por la interfaz adecuada. Esta función varía según la posición del router en el dominio MPLS, es decir, además de intercambiar, puede retirar o adicionar etiquetas.

Las funciones que realizan los nodos tienen nombres bien definidos y pueden ser realizadas sobre una etiqueta o una pila de ellas.

SWAP: Intercambiar etiqueta/s.

PUSH: Añadir etiqueta/s.

POP: Retirar etiqueta/s.

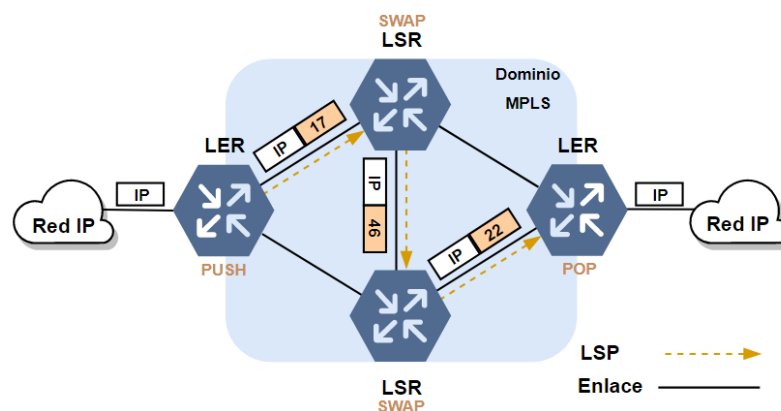


Fig 1.1 Nodos MPLS y funciones

1.1.2. Plano de control

De forma general, MPLS utiliza los prefijos anunciados por los protocolos de enrutamiento IGP/EGP y otros parámetros para la asignación de etiquetas. Los nodos MPLS registran e intercambian sus etiquetas a través de sesiones TCP para formar trayectorias LSP. Los procesos descritos anteriormente son realizados por el protocolo LDP.

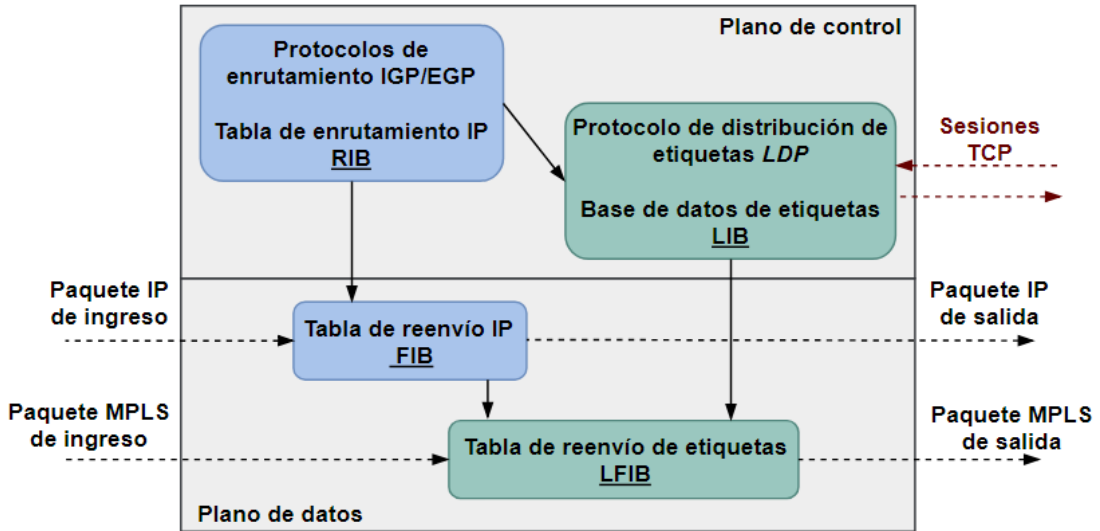


Fig 1.2 Plano de control MPLS

1.1.3. Cabecera MPLS

El conjunto de etiquetas que un nodo procesa se denomina cabecera MPLS. Cada etiqueta está compuesta por cuatro campos: Label, EXP, S y TTL.

Label: Campo de 20 bits; es el valor de la etiqueta.

EXP: Campo de 3 bits; define la QoS asignada a un conjunto de paquetes que son tratados del mismo modo por el nodo (FEC por sus siglas en inglés, *Forward Equivalence Class*).

S: Campo de 1 bit; es el indicador de pila. Cuando el valor de S es 1, indica el fin de ella.

TTL: Campo de 8 bits; realiza la misma función que el TTL IP. Si el valor del TTL es 0, el paquete es descartado, de otro modo, su valor se decrementa en 1 en cada paso por un LSR.

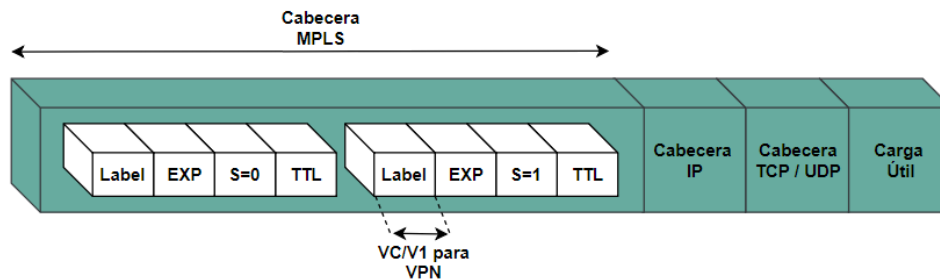


Fig 1.3 Cabecera MPLS

Es importante mencionar que para servicios VPN, la etiqueta interna es siempre la etiqueta de túnel y no se procesa por nodos intermedios LSR.

1.1.4. MPLS VPN [2]

De forma conceptual, las redes privadas virtuales permiten el acceso y compartición de información a través de túneles lógicos configurados sobre una red compartida. Según la participación del proveedor en el enrutamiento del cliente y el número de sitios que conecta, las MPLS VPN pueden clasificarse según se muestra en la figura 1.4:

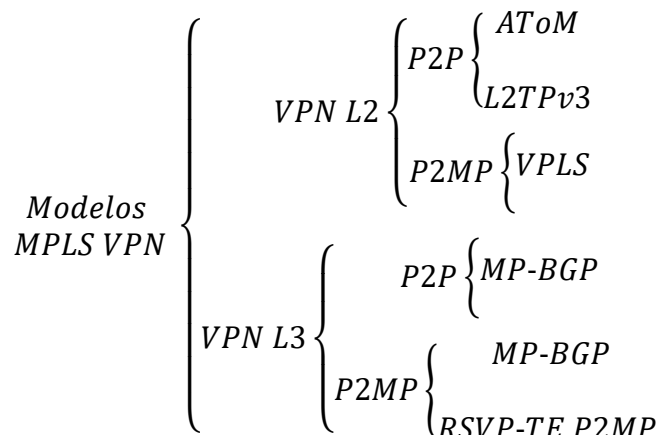


Fig. 1.4 Clasificación de los servicios MPLS VPN

A. VPN L2

En este modelo de VPN no existe compartición de rutas entre el cliente y el proveedor. Desde la perspectiva del cliente, la red del proveedor actúa como un switch, por lo tanto, es importante considerar la propagación de PDU's de nivel 2.

La asignación de etiquetas de túnel VC puede ser realizada a través de los protocolos LDP o MP-BGP.

LDP: Ambos LER asignan una etiqueta VC a la VC ID previamente configurada y envían un mensaje de asignación de etiqueta a su par a través de una sesión LDP. El LER receptor mapea este mensaje con el VC ID local y si hay coincidencia utiliza la etiqueta codificada en el mensaje de asignación para reenviar los paquetes hacia su par.

MP-BGP: Es una extensión del protocolo BGP y además de anunciar prefijos IPv4 permite anunciar prefijos IPv6, l2vpn, vpnv4, por mencionar algunos. En el contexto de las VPN L2, MP-BGP es utilizado para anunciar prefijos l2vpn e intercambiar etiquetas VC con sus pares.

En ambos casos el protocolo LDP es siempre el encargado de asignar la etiqueta externa de la cabecera MPLS.

La figura 1.5 muestra lo expuesto; el Cliente A distribuye la etiqueta de túnel VC a través de MP-BGP mientras que el Cliente B lo hace a través de LDP.

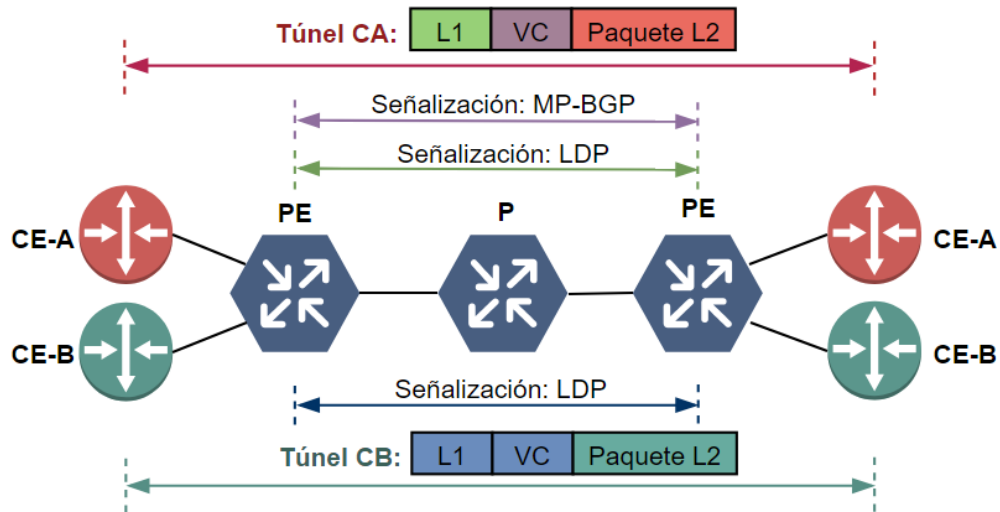


Fig. 1.5 Intercambio de VC con LDP y MP-BGP

La independencia de MPLS con tecnologías subyacentes permite desplegar servicios *Any-to-Any* entre tecnologías legacy como ATM o Frame Relay a través de túneles L2. [3]

B. VPN L3

En este modelo de VPN es necesario el intercambio de información de enrutamiento entre el cliente y el proveedor. Las rutas entregadas por el cliente son anunciadas entre los nodos LER a través de sesiones MP-BGP y su privacidad es apoyada por tablas de enrutamiento virtuales denominadas VRF. Cada VRF posee parámetros relevantes que deben ser configurados:

Route Distinguisher: Valor de 64 bits. Se añade al prefijo de red y juntos forman una dirección de 96 bits denominada *vpn4*. Permite el solapamiento de direcciones IPv4.

Route Target: Valor de 64 bits. Se utiliza para identificar los prefijos *vpn4* que cada LER aprende o publica a través de MP-BGP.

VPN ID: Parámetro usado para identificar la pertenencia de una VRF a un árbol MP2M2 compartido. Este parámetro es configurado para el enrutamiento multicast en conjunto con el árbol de distribución de multidifusión MDT.

Al igual que en VPN L2, MP-BGP se utiliza para anunciar prefijos *vpn4* e intercambiar etiquetas de túnel V1. El protocolo LDP se encarga de asignar la etiqueta externa de la cabecera MPLS.

Es necesario recalcar que la compartición de rutas entre el cliente y el proveedor puede ser a través de rutas estáticas, enrutamiento dinámico o BGP.

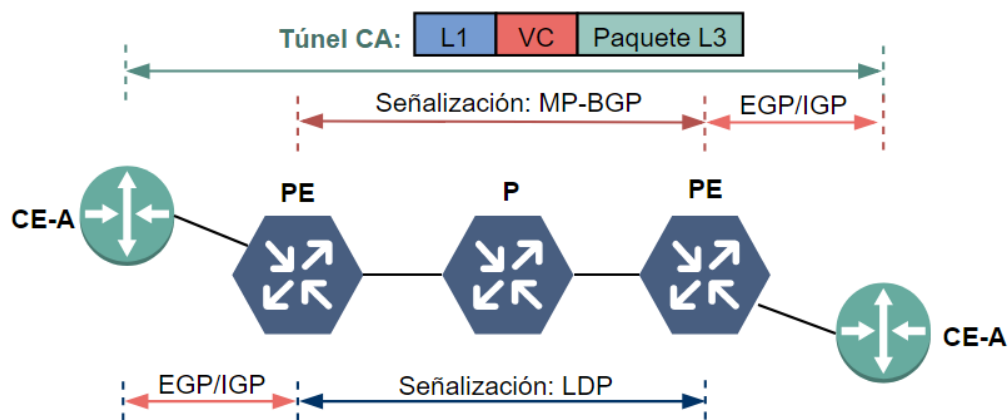


Fig. 1.6 Señalización de etiquetas en servicios MPLS VPN L3

1.1.5. Calidad de servicio – QoS

QoS es un mecanismo que permite priorizar aplicaciones, usuarios o flujos de datos a través de algunos procesos sistemáticos. Estos procesos comprenden la clasificación y programación de políticas sobre los paquetes:

Clasificación: Se marcan y clasifican los paquetes en función del tipo de tráfico que estos transportan. Las marcas se pueden establecer en varios niveles según el modelo TCP/IP utilizando los campos COS, DSCP y EXP de los *header* correspondientes.

Tabla 1.1: Marcas de QoS en función del tipo de tráfico

<i>Aplicación</i>	<i>IPP</i>	<i>PHB</i>	<i>DSCP</i>	<i>COS/EXP</i>
<i>Enrutamiento IP</i>	6	CS6	48	6
<i>Voz</i>	5	EF	46	5
<i>Video interactivo</i>	4	AF41	34	4
<i>Streaming de video</i>	4	CS4	32	4
<i>Datos de misión crítica</i>	3	-	25	3
<i>Señalización de llamadas</i>	3	AF31/CS3	26/24	3
<i>Datos transaccionales</i>	2	AF21	18	2
<i>Gestión de red</i>	2	CS2	16	2
<i>Datos masivos</i>	1	AF11	10	1
<i>Scavenger</i>	1	CS1	8	1
<i>Mejor esfuerzo</i>	0	0	0	0

Configuración de políticas: Se establecen políticas como ancho de banda, límite de cola, etc., para las clases creadas y en función de esas políticas, las clases se ponderan y ajustan a un modelo de colas como CBWFQ o CBWFQ-LLC. Esta ponderación o peso define la prioridad de cada clase en el modelo. Las políticas de QoS pueden anidarse y post-configuración, se aplican a las interfaces involucradas.

1.2. Métricas de desempeño en redes IP [4]

De forma general, las métricas de desempeño permiten evaluar el estado de una red, sus servicios y las políticas de QoS configuradas sobre ella. En redes IP, estas métricas están estandarizadas por el IETF y la ITU-T.

1.2.1. Métricas IETF

Las métricas IETF especificadas en los RFCs 2678, 2679, 2680, 2681 y 3393 se agrupan en términos de disponibilidad, pérdida, retardo y utilización.

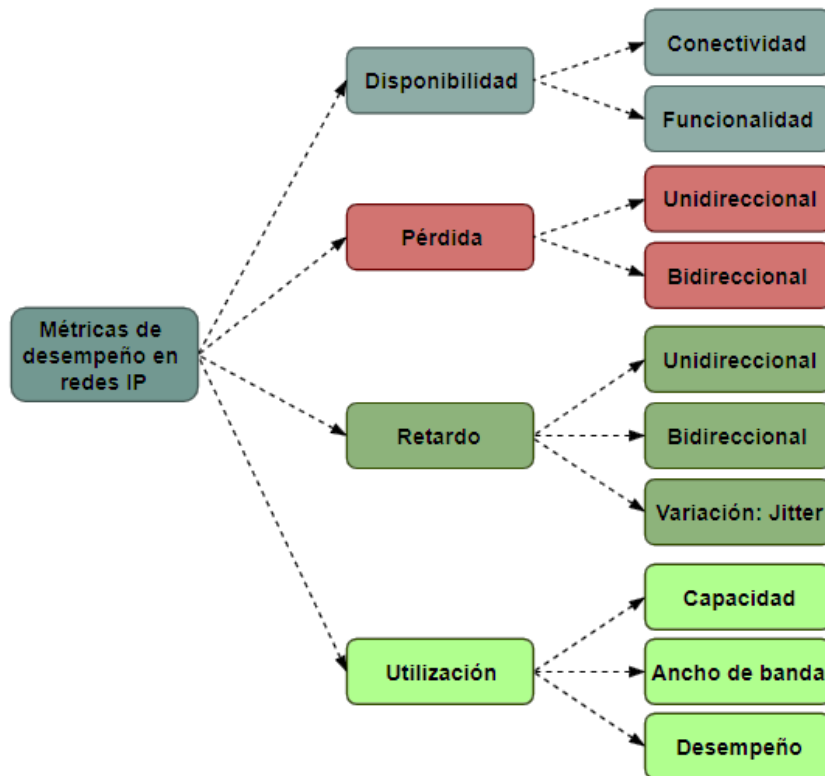


Fig. 1.7 Métricas de desempeño en redes IP

Disponibilidad: Se expresa en términos porcentuales y representa el tiempo que un servicio puede ser accedido en una ventana temporal acotada.

$$D = \frac{UP_{time}}{UP_{time} + DOWN_{time}}$$

Conectividad: Conectividad física de los nodos de red involucrados en el servicio.

Funcionalidad: Calidad del servicio provisto.

Pérdida de paquetes: Cantidad de paquetes perdidos durante una sesión. Pueden medirse en una o ambas direcciones.

Retardo: Tiempo que tarda un paquete en viajar entre su origen y destino. La variación de esta métrica se conoce como jitter.

Utilización: Rendimiento del servicio/red en función de su capacidad máxima. Incluye errores de nodo y protocolo.

1.2.2. Métricas ITU-T

Las métricas ITU-T permiten estimar la calidad y degradación de un servicio de voz en redes IP.

E-Model (MOS, Mean Opinion Score): Valor numérico entre 1 y 5. Indica la calidad de una sesión de voz en función de varios parámetros como *packet loss*, *códec*, etc. El documento ITU-T G.107 describe a detalle como estimarla.

Impairment / Calculated Planning Impairment Factor (ICPIF): Valor numérico entre 5 y 55. Indica el nivel de degradación de una sesión de voz y se obtiene mediante la adición de varios elementos de degradación. El documento ITU-T G.113 describe a detalle como estimarla.

1.3. Técnicas de monitoreo activo y pasivo [4]

Las técnicas de monitoreo activo inyectan tráfico en la red y al final de su ejecución despliegan reportes que incluyen métricas más complejas como índices de calidad de llamada o video. En contraparte, las técnicas de monitoreo pasivo no inyectan tráfico a la red y utilizan sondas o gestores de monitoreo SNMP para recopilar información. El cálculo de métricas más complejas como el MOS e ICPIF se realiza a través de técnicas de modelación.

1.4. Redes Neuronales Artificiales Feed-forward [5] [6]

Son modelos computacionales no paramétricos utilizados para modelar, predecir o clasificar fenómenos. Están compuestos por nodos interconectados en capas y en su forma más básica necesitan al menos una capa intermedia para “romper” la linealidad entre las capas de entrada y salida. Los nodos y enlaces que los interconectan se denominan neuronas y pesos respectivamente.

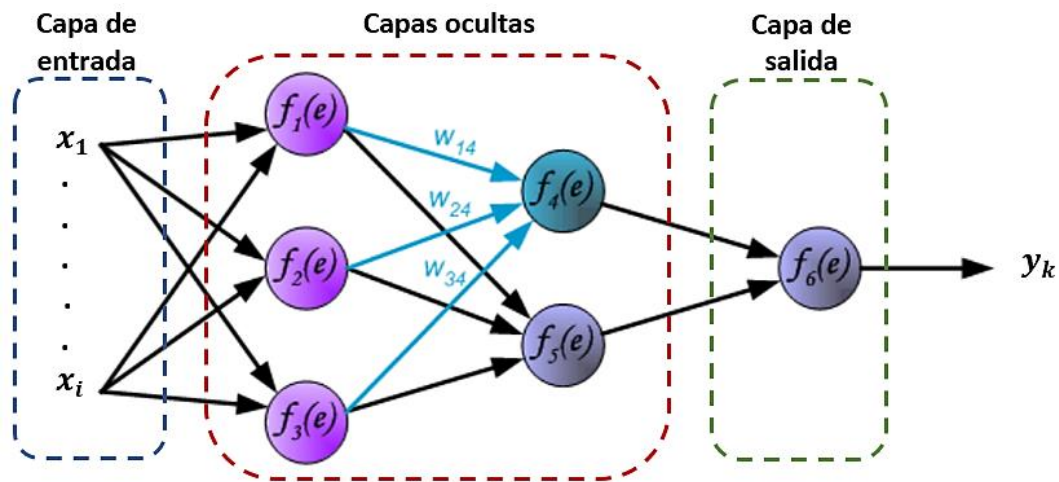


Fig. 1.8 Representación gráfica de una red neuronal artificial

Cada neurona contiene una función de activación, un valor de umbral y un bias. El valor de umbral es el valor mínimo que debe tener una entrada para activar la neurona y la función de activación es el modelo matemático aplicado sobre la entrada para obtener el valor que se compara con el umbral. El bias permite que el umbral se desplace hacia arriba o hacia abajo. Existen varios tipos de funciones de activación y se escogen según la naturaleza del fenómeno que se desea modelar; el

más común es el sigmoide. La transformación no lineal que realiza una neurona sobre los parámetros de entrada se rige por la ecuación 1.2.

$$v_k = \sum_{j=1}^m (w_{kj} * x_j) + b_k \quad (1.1)$$

$$y_k = f_k(v_k) = \frac{1}{1 + e^{-v_k}} \quad (1.2)$$

donde y_k es la salida de la neurona, w_{kj} el j -ésimo peso de entrada, x_j la j -ésima entrada, b_k el bias y f_k la función de activación, en este caso, la función sigmoide.

El proceso de aprendizaje de las redes neuronales se basa en un algoritmo denominado *backpropagation* y su objeto es encontrar una combinación adecuada de pesos que minimicen el error de ajuste. En un inicio, se calcula la salida de la red utilizando pesos aleatorios. Esa salida se compara con los valores esperados y se obtiene el error de ajuste δ_k . Este error se propaga hacia atrás para calcular el error de los nodos restantes y los nuevos pesos según las ecuaciones 1.4 y 1.5.

$$\delta_k = \sum_{k=1}^n (y_k - t_k)^2 \quad (1.3)$$

$$\delta_k^L = \sum_{k=1}^{m^L} \cdot \sum_{j=1}^{m^{L+1}} (w_{kj}^L * \delta_j^{L+1}) \quad (1.4)$$

$$w'_{kj} = w_{kj} + \rho \delta_k^L \frac{\partial f_k(v_k)}{dv_k} x_j \quad (1.5)$$

Donde δ_k es el error de ajuste, t_k el valor esperado, δ_k^L el error de los nodos de las capas ocultas, δ_j^{L+1} el error de la j -ésima neurona de la capa posterior, w_{kj}^L el peso dirigido a la j -ésima neurona de la capa posterior, w'_{kj} el peso ajustado, $\frac{\partial f_k(v_k)}{dv_k}$ el gradiente de la función de activación de la neurona y ρ el tamaño del paso de aprendizaje.

Este proceso es iterativo y los pesos se actualizan en dirección opuesta al gradiente hasta encontrar un mínimo. El factor de aprendizaje ρ es ajustable y la idea es combinarlo con un valor denominado *momentum* para evitar estancarse en mínimos locales no significativos.

1.5. Test de Anderson-Darling [7]

Es una prueba estadística no paramétrica que permite determinar si un conjunto de datos se ajusta a una distribución de probabilidad dada. La prueba calcula el estadístico A según la ecuación 1.6 y lo aproxima a una normal según la ecuación 1.7. La hipótesis nula H_0 se acepta o descarta tras comparar Z con los puntos críticos de una normal Z_α .

$$A = -n - \frac{1}{n} \sum_{i=1}^n [2i - 1] [\ln(p_{(i)}) + \ln(1 - p_{(n-i+1)})] \quad (1.6)$$

$$Z = A \left(1.0 + \frac{0.75}{n} + \frac{2.25}{n^2} \right) \quad (1.7)$$

$$p_{(i)} = \Phi \left(\frac{[x_{(i)} - \tilde{x}]}{\sigma} \right) \quad (1.8)$$

donde Φ es la CDF de la distribución, en este caso distribución normal estándar, \tilde{x} la media y σ la desviación estándar del conjunto de datos.

1.6. Bootstrapping [8] [9]

Es un método de simulación que permite realizar inferencias sobre poblaciones a partir de muestras aleatorias únicas. De forma general, el conjunto de datos es re-muestreado B veces, en cada iteración se estima y almacena el estadístico de interés. El vector de estadísticos resultante se utiliza para varios fines, entre ellos, estimar EDF, intervalos de confianza, bias, etc. Bootstrapping no realiza supuestos de distribuciones paramétricas, utiliza re-muestreo con reemplazo y funciona bien con muestras medianas y grandes.

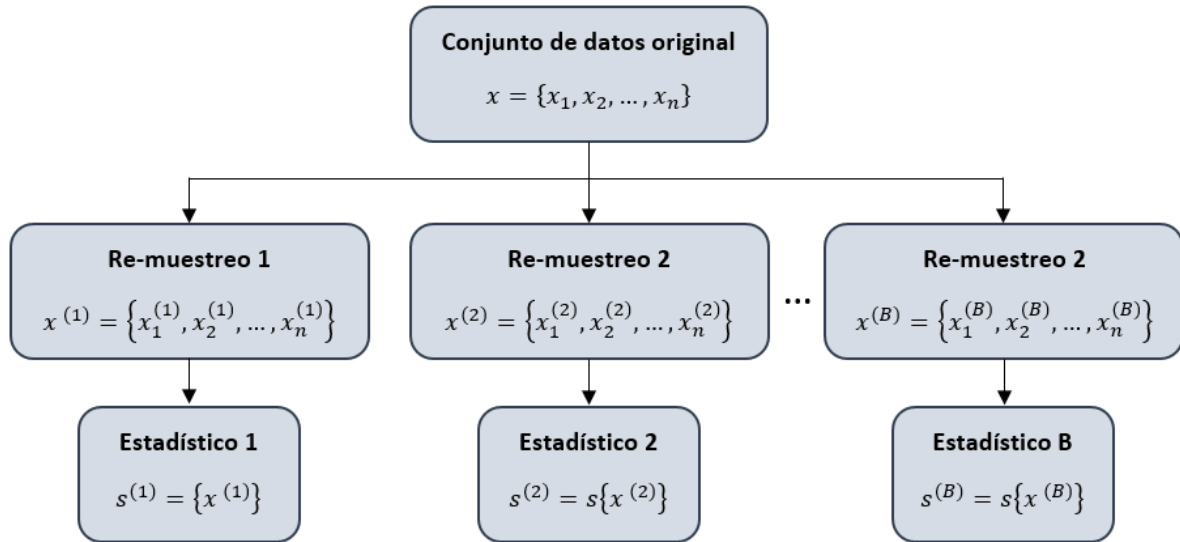


Fig. 1.9 Representación general de Bootstrapping

1.7. Transformada de Box-Cox [10]

Es una familia de transformaciones de potencia que permite dar normalidad a un conjunto de datos. La transformación se rige por la ecuación 1.9.

$$Z(\lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{Si } \lambda \neq 0 \\ \log(x) & \text{Si } \lambda = 0 \end{cases} \quad (1.9)$$

El parámetro λ se estima maximizando el valor de la función de probabilidad descrita en 1.10; x es el conjunto de datos.

$$f(x, \lambda) = -\frac{N}{2} \ln \left(\sum_{i=0}^{N-1} \left(\frac{x_i(\lambda) - \bar{x}(\lambda)}{N} \right)^2 \right) + (\lambda - 1) \sum_{i=0}^{N-1} \ln(x_i) \quad (1.10)$$

$$\bar{x}(\lambda) = \frac{1}{N} \sum_{i=0}^{N-1} x_i(\lambda) \quad (1.11)$$

1.8. Prueba de U Mann-Whitney [11]

Es una prueba estadística no paramétrica que permite determinar la existencia de diferencias significativas entre dos muestras en función de sus medianas. La prueba calcula el estadístico U según la ecuación 1.14 y lo aproxima a una normal según la ecuación 1.15. La hipótesis nula H_0 se acepta o descarta tras comparar Z con los puntos críticos de una normal Z_α .

$$U_1 = n_1 * n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad (1.12)$$

$$U_2 = n_1 * n_2 + \frac{n_2(n_2 + 1)}{2} - R_2 \quad (1.13)$$

$$U = \min\{U_1, U_2\} \quad (1.14)$$

$$Z = \frac{U - \left(\frac{n_1 * n_2}{2}\right)}{\sqrt{\frac{n_1 * n_2 (n_1 + n_2 + 1)}{12}}} \equiv N(0,1) \quad (1.15)$$

Los valores R_1 y R_2 corresponden a la suma de los rangos de las muestras n_1 y n_2 respectivamente.

1.9. R Statistics [12]

R es un lenguaje y entorno de análisis estadístico que proporciona una amplia variedad de herramientas de estadística clásica, modelado lineal y no lineal, pruebas estadísticas paramétricas y no paramétricas, análisis de series de tiempo, clasificación, agrupamiento, etc. Es altamente extensible y posee integración con otras plataformas como TensorFlow.

Capítulo 2

Metodología

En este capítulo se describen los procesos de diseño, implementación y simulación de los escenarios. La figura 2.1 muestra una visión general de cada proceso.

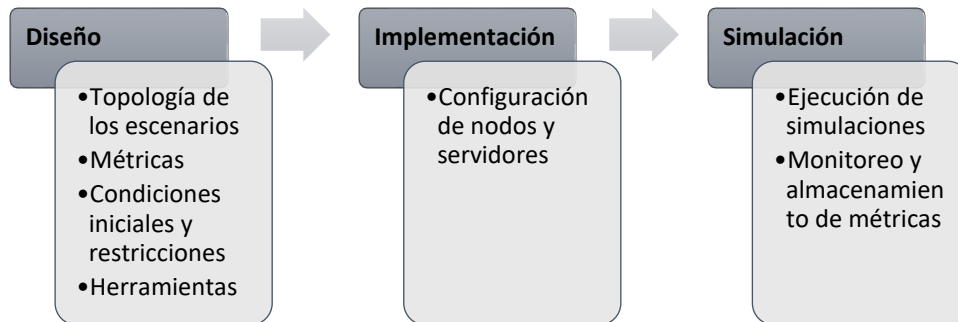


Fig 2.1 Metodología de la investigación

2.1. Diseño

2.1.1. Topología de los escenarios

La topología de los escenarios debe garantizar que el desempeño de los servicios MPLS VPN se ponga a prueba bajo las mismas condiciones. En base a ese principio se plantea: nodos de borde y políticas de QoS comunes para los servicios MPLS VPN, nodos de cliente de igual capacidad, además, se prescinde de enlaces redundantes en el *backbone* para garantizar cargas iguales de CPU y concentrar el tráfico en un único enlace; este último es útil para reducir el coste computacional durante la simulación.

Otras consideraciones:

- a. Clientes con acceso a servicios MPLS VPN e internet.
- b. Capacidad de inyección de tráfico de *backbone*.
- c. Capacidad de monitoreo y almacenamiento de métricas.
- d. Servidor de sincronización.
- e. *Backbone* sin visibilidad desde internet.

Los escenarios para tráfico unicast y multicast poseen las mismas características, pero son tratados como experimentos independientes en la etapa de simulación. La figura 2.2 muestra la topología planteada.

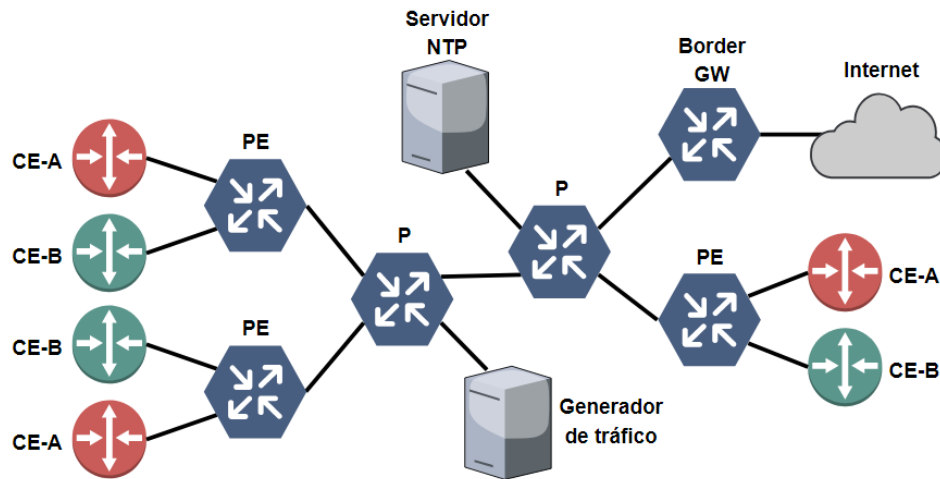


Fig 2.2 Topología de los escenarios unicast y multicast

2.1.2. Métricas de desempeño

Las métricas utilizadas para comparar el desempeño de los servicios MPLS VPN L2 y L3 son: *packet loss*, *one-way delay*, *one-way jitter*, *round-trip-time*, *ICPIF* y *MOS*.

2.1.3. Condiciones Iniciales

Un error común en las pruebas de desempeño de protocolos, servicios o políticas; es la omisión del tráfico de control y usuario que atraviesa una red operativa. Esto ocasiona que las métricas obtenidas en los análisis estén sobrevaloradas y las proyecciones realizadas a partir de ellas fallen en condiciones de carga.

Para evitar aquello, se modela el tráfico de ocupación del enlace entre el proveedor de servicios VTR y el PIT Entel¹ para incorporarlo como tráfico de backbone en los escenarios y evitar métricas sobrevaloradas post-simulación. Las muestras disponibles corresponden a 11 meses de monitoreo, su comportamiento y función de distribución de probabilidad se observa en la figura 2.3.

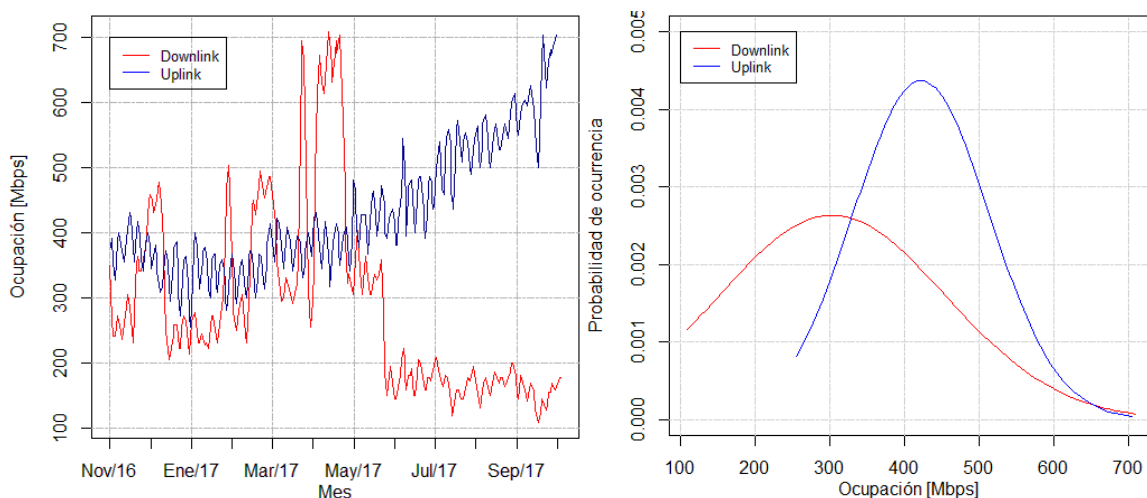


Fig 2.3 Comportamiento y PDF del tráfico VTR - Entel

¹ Información obtenida de: <http://www.pitentel.cl/estadistica.php?enlace=vtr1&periodo=anual&tresD=0>

La PDF indica que las muestras no siguen una distribución normal, por lo tanto, utilizar estadísticos como la media y la desviación estándar para modelar su comportamiento no proporciona una solución adecuada, la figura 2.4 es un ejemplo de ello; por este motivo se utiliza redes neuronales artificiales.

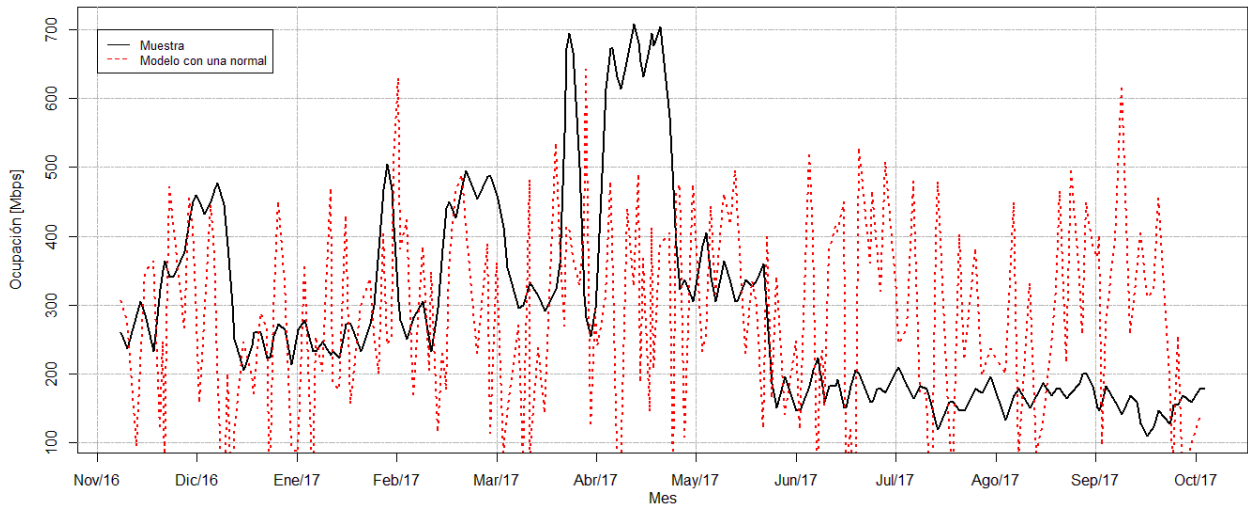


Fig 2.4 Caracterización del tráfico con una distribución normal

Las variables de predicción x_i de la red neuronal se obtienen tras aplicar la función de autocorrelación parcial (PACF) sobre las muestras de tráfico y_i . La figura 2.5 exhibe la PACF de las muestras de tráfico *uplink* y *downlink*. El número de variables de predicción de cada modelo es proporcional al número de retardos sobre el umbral proporcionado por la PACF, en este caso, cinco para el tráfico *uplink* y dos para el tráfico *downlink*. Todo este conjunto de variables compone el set de entrenamiento y se estandariza para evitar que las neuronas se activen con cada valor de entrada.

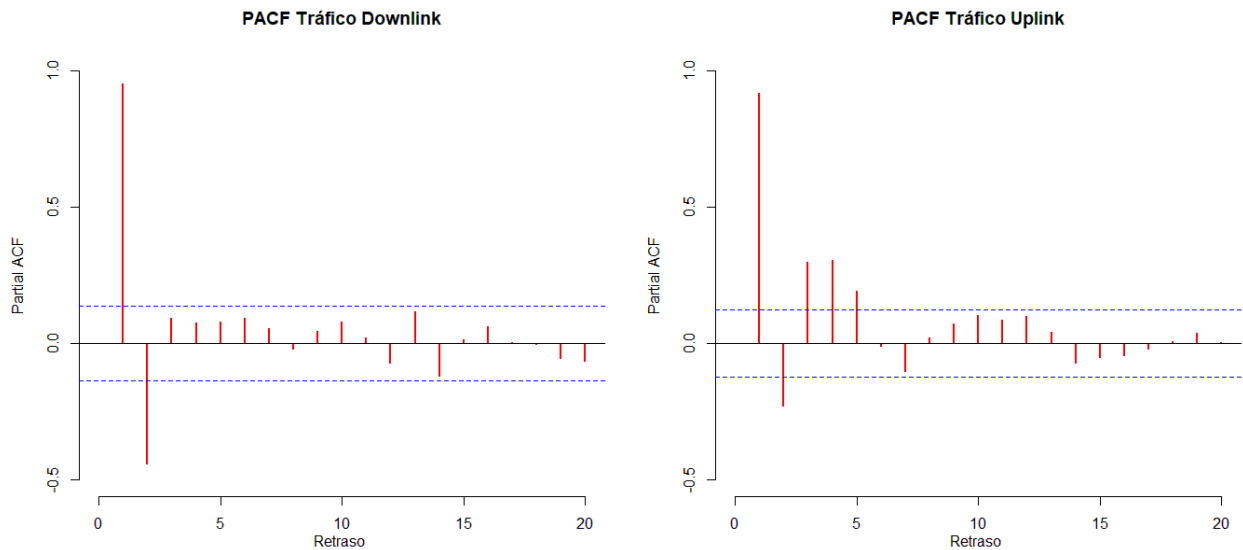


Fig. 2.5 PACF de las muestras de tráfico

La figura 2.6 indica el proceso de caracterización del tráfico; se utiliza el entorno de programación R y las librerías *caret* y *nnet* para construir los modelos. Los algoritmos se adjuntan en el anexo A.

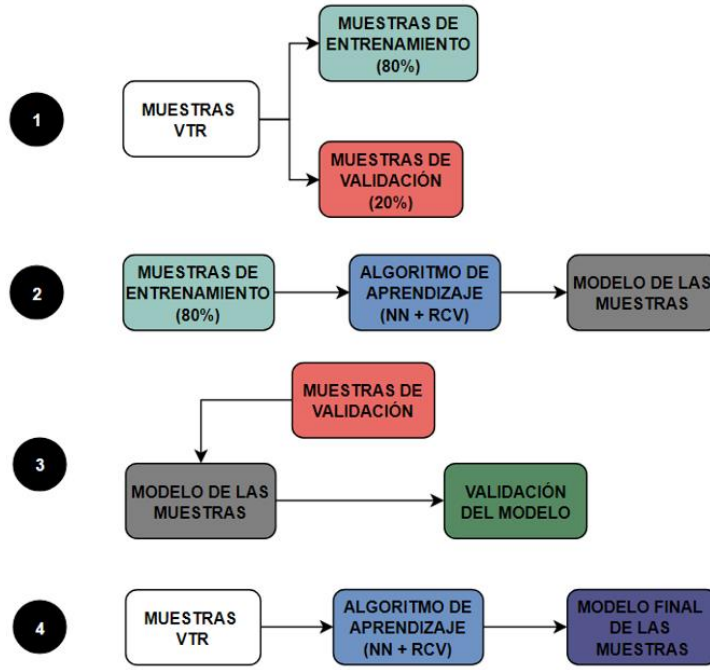


Fig. 2.6 Proceso de caracterización del tráfico

El modelo matemático de las muestras de tráfico *uplink* y *downlink* se rige por las ecuaciones 2.1 y 2.2 respectivamente.

$$h_1 = \frac{1}{1 + e^{-((2.77*x_1 - 0.95*x_2 + 1.36*x_3 - 1.57*x_4 + 0.21*x_5) - 2.24)}}$$

$$h_2 = \frac{1}{1 + e^{-((0.74*x_1 + 0.83*x_2 + 0.84*x_3 - 1.96*x_4 - 0.49*x_5) - 0.24)}}$$

$$h_3 = \frac{1}{1 + e^{-((2.50*x_1 - 1.85*x_2 - 0.36*x_3 + 1.23*x_4 + 0.81*x_5) - 1.56)}}$$

$$y = \frac{1}{1 + e^{-((1.38*h_1 - 0.80*h_2 + 0.56*h_3) + 0.21)}} \quad (2.1)$$

$$h_1 = \frac{1}{1 + e^{-((9.28*x_1 - 4.11*x_2) - 3.66)}}$$

$$h_2 = \frac{1}{1 + e^{-((-0.65*x_1 - 1.70*x_2) - 1.52)}}$$

$$h_3 = \frac{1}{1 + e^{-((2.62*x_1 - 4.20*x_2) - 1.19)}}$$

$$y = \frac{1}{1 + e^{-((0.79*h_1 + 1.51*h_2 + 0.63*h_3) - 0.37)}} \quad (2.2)$$

El error porcentual absoluto medio (MAPE) de los modelos de tráfico *uplink* y *downlink* es de 2.67% y 3.02% respectivamente. Las figuras 2.7 y 2.8 exhiben el ajuste del tráfico.

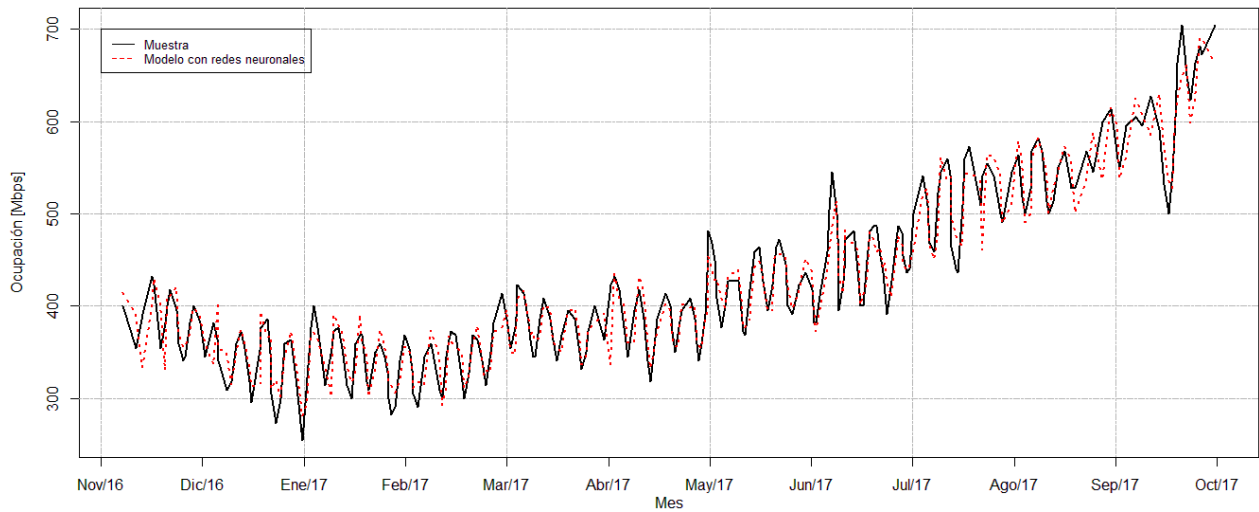


Fig. 2.7 Muestra y modelo del tráfico *uplink*

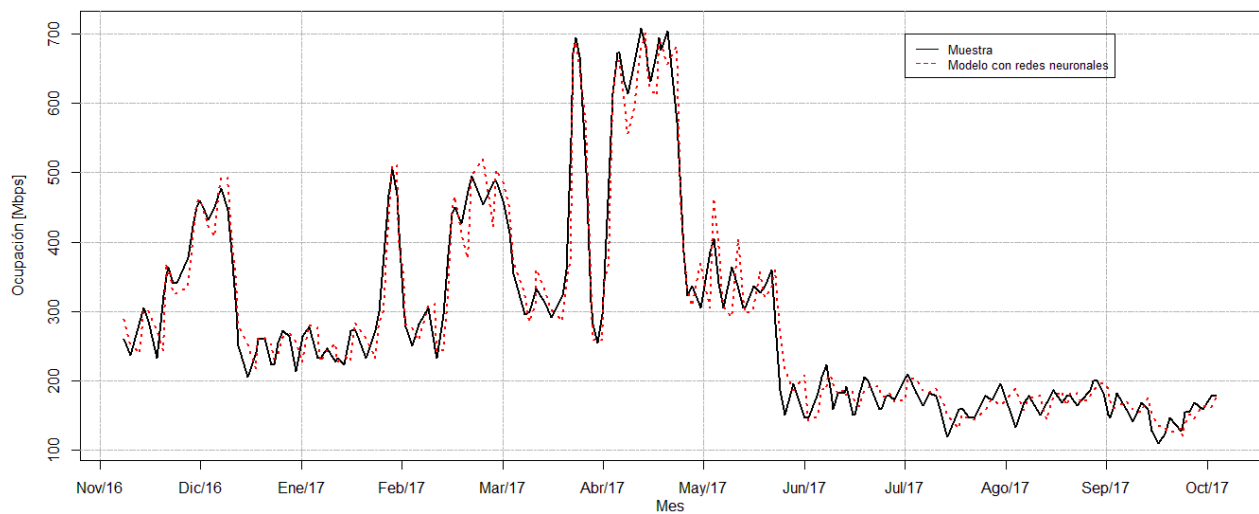


Fig. 2.8 Muestra y modelo del tráfico *downlink*

El objeto de todo esto es inyectar el tráfico modelado en los escenarios durante la simulación y así obtener las métricas descritas en el punto 2.1.2 para comparar el desempeño de los servicios MPLS VPN L2 y L3.

2.1.4. Herramientas

Para desplegar los escenarios se utiliza el simulador gráfico GNS3². Es de distribución libre y su capacidad de integración con varias herramientas de *networking* permite obtener el control total del experimento en un entorno común. La tabla 2.1 detalla las herramientas utilizadas en este trabajo.

² Sitio Web: <https://www.gns3.com/>

Tabla 2.1 Características de las herramientas

Requisito	Herramienta
Nodos de cliente y <i>backbone</i>	Router Cisco 7200
Servidor de sincronización	Open NTPD sobre VM Alpine
Generador de tráfico	Iperf sobre VM Alpine y Windows
Servidor de almacenamiento	WinAgents TFTP Server
Monitoreo de métricas	IP SLA y Wireshark

2.1.5. Restricciones

El rendimiento de los router virtualizados con Dynamips se torna inestable sobre los 9 Mbps y cualquier ráfaga de tráfico que supere ese umbral proporcionará métricas erróneas durante la simulación. Esta es una limitación propia de Dynamips y no tiene relación con las capacidades del equipo anfitrión. La transformada de BoxCox aplicada a la distribución de estadísticos obtenidos con Bootstrapping, permite estimar la media “poblacional” escalada (tráfico que se incorpora en la simulación) y sus intervalos de confianza (C.I.). El proceso se observa en la siguiente figura.

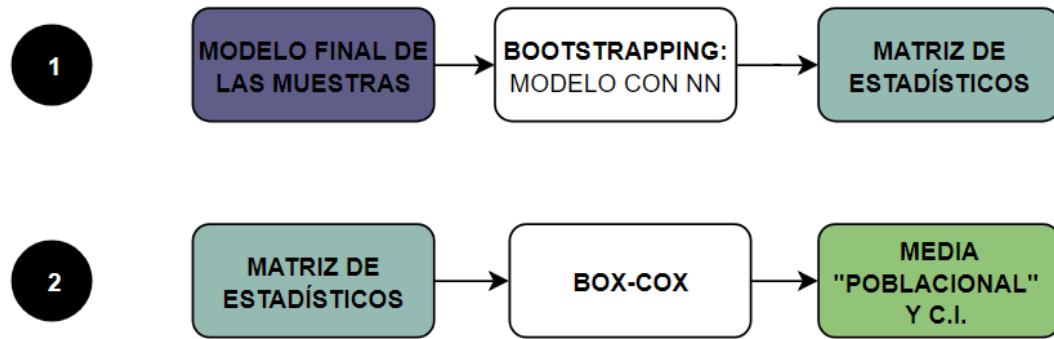


Fig. 2.9 Proceso para la estimación de la EDF y los intervalos de confianza

El coeficiente lambda escogido para escalar la matriz de estadísticos de tráfico es de 0.60905; la transformación se realiza conforme a la ecuación 1.9. El algoritmo para estimar el tráfico de referencia se presenta a continuación:

Algoritmo 1 Algoritmo para calcular el tráfico de referencia

Requiere: $x_n, B, s(x_n)$

- 1: for i in 1:B
 - 2: Calcular x_n^* = re-muestreo con reemplazo de x_n
 - 3: Calcular $s(x_n^*)$
 - 4: Almacenar $v[i,1] = s(x_n^*)$
 - 5: Calcular media de $v[,1]$
-

Tras realizar ambos procesos, los valores de tráfico *uplink* y *downlink* son 3 Mbps y 2.7 Mbps respectivamente.

2.2. Implementación

La configuración de MPLS y sus servicios necesita la ejecución de un IGP como prerequisite. En este trabajo se asume que los nodos de *backbone* ejecutan un protocolo de enrutamiento y son alcanzables entre sí. Además se asume que las sesiones BGP entre los *peer* están configuradas.

2.2.1. Configuración MPLS

La configuración se realiza en todas las interfaces que componen el dominio MPLS; los comandos se detallan en la tabla 2.2.

Tabla 2.2 Configuración de MPLS

<i>Sistema Operativo</i>	<i>Nodos</i>	<i>Configuración</i>
<i>IOS</i>	<i>P y PE</i>	<i>interface <if></i> <i>mpls ip</i> <i>!</i>

2.2.2. Unicast

A. Configuración MPLS VPN L3

El servicio se habilita solamente en los nodos de borde PE, su configuración se detalla en la tabla 2.3.

Tabla 2.3 Configuración de MPLS VPN L3

<i>Sistema Operativo</i>	<i>Nodos</i>	<i>Configuración</i>
<i>IOS</i>	<i>PE</i>	<i>vrf definition <nombre></i> <i>rd <id></i> <i>address-family ipv4</i> <i>route-target export <id></i> <i>route-target import <id></i> <i>exit</i> <i>!</i> <i>interface <if-PE-CE></i> <i>vrf forwarding <nombre></i> <i>!</i> <i>router bgp <AS></i> <i>address-family vpnv4</i> <i>neighbor <peer> activate</i> <i>neighbor <peer> send-community extended</i> <i>exit</i> <i>!</i> <i>address-family ipv4 vrf <nombre></i> <i>redistribute o network</i>

Este modelo de VPN requiere el intercambio de información de nivel 3 entre el proveedor del servicio y el cliente.

B. Configuración MPLS VPN L2

Se opta por desplegar un servicio de tipo EoMPLS. Su configuración se detalla en la tabla 2.4.

Tabla 2.4 Configuración de MPLS VPN L2

<i>Sistema Operativo</i>	<i>Nodos</i>	<i>Configuración</i>
<i>IOS</i>	<i>PE</i>	<i>interface <if-PE-CE> xconnect <peer> <vcid> encapsulation mpls !</i>

El parámetro *peer* es la dirección IP del par PE y el parámetro *vcid* es el identificador de etiqueta VC que debe ser único para cada cliente.

2.2.3. Multicast

A. Configuración MPLS VPN L3

El servicio se habilita en los nodos de borde y cliente; su configuración se detalla en la tabla 2.5.

Tabla 2.5 Configuración de MPLS VPN L3 multicast

<i>Sistema Operativo</i>	<i>Nodos</i>	<i>Configuración</i>
<i>IOS</i>	<i>PE</i>	<i>vrf definition <nombre> rd <id> vpn id <id> address-family ipv4 route-target export <id> route-target import <id> ! address-family ipv4 mdt preference mldp mdt default mpls mldp <nodo-raiz> mdt data mpls mldp <número de árboles de datos P2MP> exit-address-family ! interface <if-PE-CE> vrf forwarding <nombre> ip pim passive ! router bgp <AS> address-family vpnv4 neighbor <peer> activate neighbor <peer> send-community extended exit ! address-family ipv4 vrf <nombre> redistribute o network ! ip multicast-routing vrf <nombre> ip pim vrf S1 rp-address <ip-RP></i>

<i>IOS</i>	<i>CE</i>	<i>interface <if-CE-PE> ip pim sparse-mode ip igmp join-group <ip-multicast></i>
------------	-----------	--

B. Configuración MPLS VPN L2

El servicio se habilita en los nodos de borde y cliente; los comandos de configuración se detallan en la tabla 2.6.

Tabla 2.6 Configuración de MPLS VPN L2 multicast

<i>Sistema Operativo</i>	<i>Nodos</i>	<i>Configuración</i>
<i>IOS</i>	<i>PE</i>	<i>interface <if> no ip address xconnect <peer> <vcid> encapsulation mpls ! interface <if> vrf forwarding <nombre> ip pim passive ! ip pim vrf S1 rp-address <ip-RP></i>
<i>IOS</i>	<i>CE</i>	<i>interface <if> ip pim sparse-mode ip igmp join-group <ip-multicast> !</i>

El acceso a internet para los clientes de los servicios VPN L2 se realiza a través de una interfaz adicional. Además, esta interfaz se aprovecha para proporcionar *Anycast RP* en multicast.

El comando *ip igmp join-group* permite prescindir de receptores multicast.

2.2.4. Configuración NTP

La configuración se realiza en la VM Alpine y en los router; el rol de los dispositivos y los comandos se detalla en la tabla 2.7.

Tabla 2.7 Configuración de NTP

<i>Nodo</i>	<i>Rol</i>	<i>Configuración</i>
<i>VM Alpine</i>	<i>Servidor</i>	<i>apk add openntp cat /etc/ntp.conf descomentar listen on *</i>
<i>Router Cisco 7200</i>	<i>Cliente</i>	<i>ntp server <ip-servidor></i>

2.2.5. Inyección del tráfico de *backbone*

El tráfico de referencia que se inyecta en el *backbone* se genera con IPERF desde VM Alpine y Windows. Para ejecutar IPERF sobre una interfaz específica en Windows se utiliza *ForceBindIP*. El esquema de conexión se muestra a continuación.

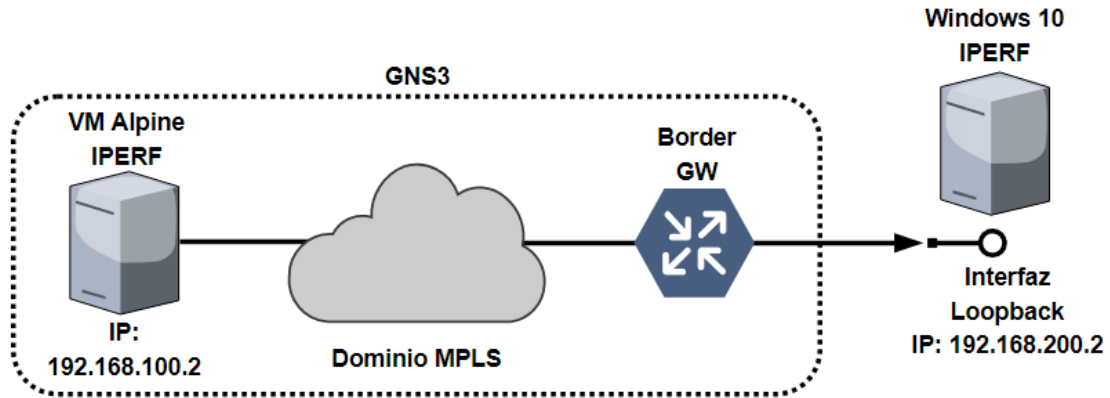


Fig. 2.10 Esquema de conexión de los generadores de tráfico

La configuración del tráfico de *backbone* y el rol de los nodos se detalla en la tabla 2.8.

Tabla 2.8 Configuración del tráfico de referencia

<i>Nodo</i>	<i>Rol</i>	<i>Configuración</i>
VM Alpine	Generador: Tráfico Uplink	<code>iperf -c <192.168.200.2> -t <43200> -u -b 3M</code>
Windows 10	Receptor: Tráfico Uplink	<code>iperf -s -i 1 -t <43200> -u -b 3M</code>
VM Alpine	Receptor: Tráfico Downlink	<code>iperf -s -i 1 -t <43200> -u -b 2.7M</code>
Windows 10	Generador: Tráfico Downlink	<code>iperf -c <192.168.100.2> -t <43200> -u -b 2.7M</code>

2.2.6. Configuración de políticas de QoS

Las políticas de QoS que se configuran garantizan que los protocolos de control LDP, BGP y NTP reciban un trato preferencial cuando exista congestión. Estas políticas no se aplican sobre los protocolos IGP porque están protegidos por el mecanismo interno de CISCO *PAK_priority* [13]. Además, se crea una política para suavizar el tráfico entre las interfaces de los nodos centrales P y evitar *peaks* que sobrepasen los 9 Mbps. La tabla 2.9 detalla lo expuesto.

Tabla 2.9 Configuración de políticas QoS

<i>S.O.</i>	<i>Nodo</i>	<i>Configuración</i>
IOS	P	<code>class-map match-all CS6</code> <code>match dscp cs6</code> <code>!</code> <code>policy-map Control</code> <code>class CS6</code>

IOS	P	<pre> bandwidth 50 policy-map Shape class class-default shape average 5000000 937500 1875000 service-policy Control ! int g1/0 service-policy output Shape </pre>
-----	---	---

Los valores de ancho de banda que se asignan en las políticas de QoS se calculan tras estimar la media poblacional con Bootstrapping. El algoritmo que se utiliza es igual al presentado en la sección 2.1.5.

2.2.7. Configuración WinAgents TFTP

El servidor TFTP es el encargado de almacenar los registros de las métricas durante la simulación y se ejecuta sobre una interfaz *loopback* en Windows 10. El esquema de conexión se muestra en la figura 2.11.

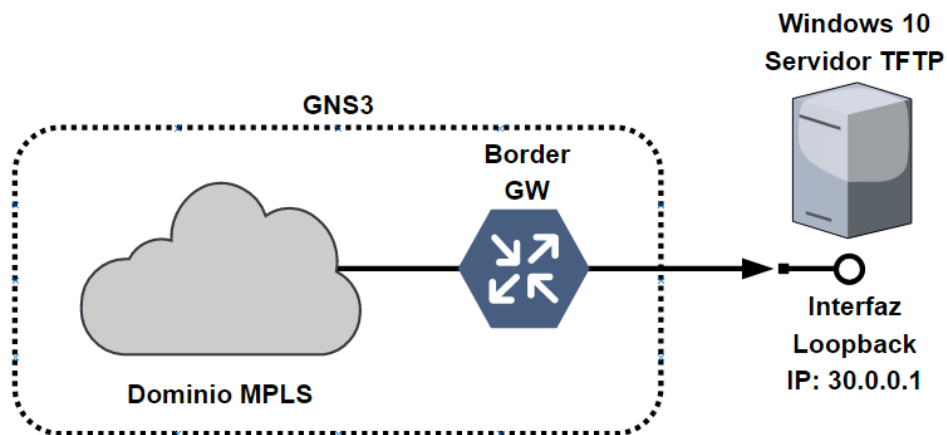


Fig. 2.11 Esquema de conexión del servidor de almacenamiento

2.3. Simulación

La etapa de simulación comprende la ejecución recurrente de dos funciones que se incluyen en el IOS de Cisco: *IP Service Level Agreements* y *Kron*. IP SLA es una función de monitoreo activo que entrega reportes compuestos por métricas como *packet loss*, *jitter*, entre otras. En este trabajo se utiliza la operación *udp-jitter* de IP SLA para obtener las métricas de desempeño entre los clientes VPN. El tiempo de vida de la operación es de 12 horas y las sesiones de voz emuladas se activan cada 60 segundos. Las sesiones generadas por IP SLA no reciben tratamiento especial en el *backbone*, por lo tanto, no hay políticas de QoS asociadas a estos flujos. Los reportes se guardan en el servidor TFTP cada 60 segundos con la ayuda de *Kron*. IP SLA se ejecuta simultáneamente para los servicios MPLS VPN L2 y L3. La figura 2.12 muestra el esquema de la simulación.

Además, se utiliza Wireshark para capturar el tráfico que fluye por el *backbone*.

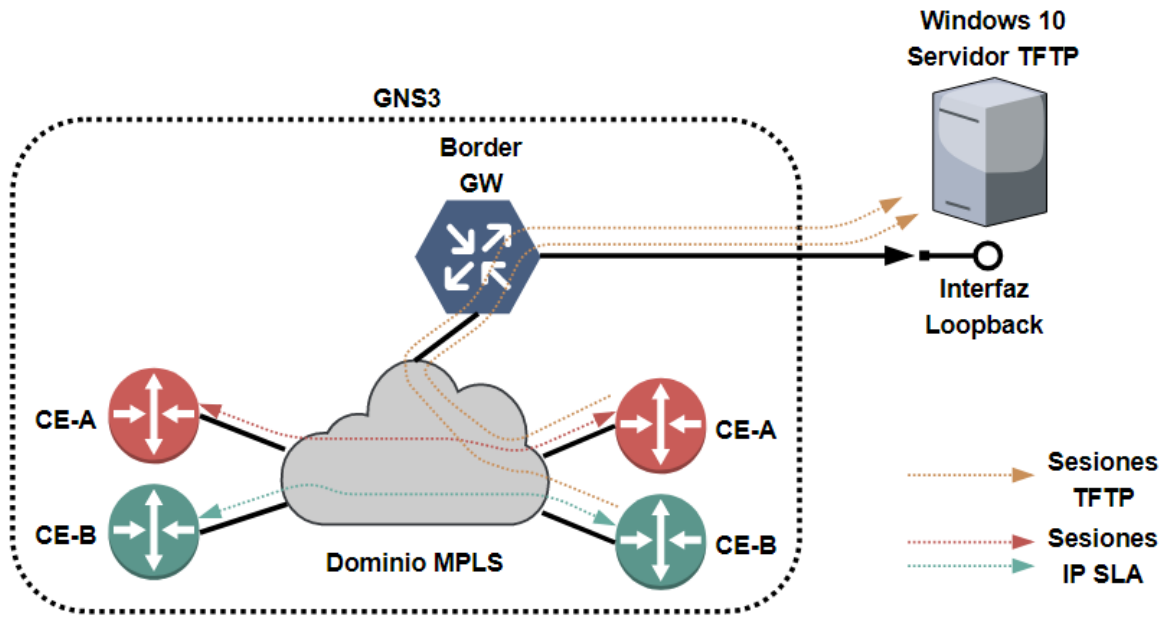


Fig. 2.12 Esquema de la simulación

Capítulo 3

Análisis de datos

En este capítulo se presenta el resultado de los test estadísticos aplicados sobre las métricas de estudio. Además, se utilizan herramientas de simulación para estimar los intervalos de confianza de la media y obtener una representación gráfica del desempeño de los servicios MPLS VPN.

3.1. Test de Normalidad

Se realizan test de normalidad sobre las métricas obtenidas post-simulación para no incurrir en el uso de recursos estadísticos inapropiados. Los resultados del test se muestran en la tabla 3.1 y 3.2.

Tabla 3.1 Test de normalidad de Anderson-Darling sobre las métricas unicast

<i>Escenario</i>	<i>Modelo</i>	<i>Métrica</i>	<i>Estadísticos</i>
<i>Unicast</i>	<i>MPLS VPN L2</i>	<i>Latencia unidireccional</i>	$A = 24.45, p\text{-value} < 2.2e-16$
		<i>Jitter unidireccional</i>	$A = 70.625, p\text{-value} < 2.2e-16$
		<i>ICPIF</i>	$A = 32.303, p\text{-value} < 2.2e-16$
		<i>MOS</i>	$A = 31.766, p\text{-value} < 2.2e-16$
		<i>Packet Loss</i>	$A = 2.6653, p\text{-value} = 1.01e-06$
		<i>RTT</i>	$A = 85.861, p\text{-value} < 2.2e-16$
<i>Unicast</i>	<i>MPLS VPN L3</i>	<i>Latencia unidireccional</i>	$A = 7.9597, p\text{-value} < 2.2e-16$
		<i>Jitter unidireccional</i>	$A = 73.556, p\text{-value} < 2.2e-16$
		<i>ICPIF</i>	$A = 34.114, p\text{-value} < 2.2e-16$
		<i>MOS</i>	$A = 32.894, p\text{-value} < 2.2e-16$
		<i>Packet Loss</i>	$A = 2.1383, p\text{-value} = 1.957e-05$
		<i>RTT</i>	$A = 16.968, p\text{-value} < 2.2e-16$

Tabla 3.2 Test de normalidad de Anderson-Darling sobre las métricas multicast

<i>Escenario</i>	<i>Modelo</i>	<i>Métrica</i>	<i>Estadísticos</i>
<i>Multicast</i>	<i>MPLS VPN L2</i>	<i>Latencia unidireccional</i>	$A = 20.674, p\text{-value} < 2.2e-16$
		<i>Jitter unidireccional</i>	$A = 84.075, p\text{-value} < 2.2e-16$
		<i>ICPIF</i>	$A = 41.353, p\text{-value} < 2.2e-16$
		<i>MOS</i>	$A = 41.396, p\text{-value} < 2.2e-16$
		<i>RTT</i>	$A = 237.21, p\text{-value} < 2.2e-16$
<i>Multicast</i>	<i>MPLS VPN L3</i>	<i>Latencia unidireccional</i>	$A = 12.242, p\text{-value} < 2.2e-16$
		<i>Jitter unidireccional</i>	$A = 76.942, p\text{-value} < 2.2e-16$
		<i>ICPIF</i>	$A = 25.496, p\text{-value} < 2.2e-16$
		<i>MOS</i>	$A = 25.277, p\text{-value} < 2.2e-16$
		<i>RTT</i>	$A = 140.2, p\text{-value} < 2.2e-16$

Nota: La simulación no produjo pérdida de paquetes en ningún servicio MPLS VPN.

La información proporcionada por los test – $p\text{-value} < 0.05$ – indica que el comportamiento de las métricas no es normal, por lo tanto, no se puede utilizar herramientas de análisis estadístico que asuman normalidad para comparar el desempeño de los servicios VPN.

3.2. Prueba U de Mann-Whitney

Se utiliza la prueba U de Mann-Whitney para determinar la existencia de diferencias significativas entre las medianas de las métricas. La prueba se ejecuta sobre pares de métricas L2 y L3 obtenidas en el mismo escenario, por ejemplo, MOS L2 unicast vs MOS L3 unicast.

Los resultados se presentan a continuación.

3.2.1. Métricas unicast

Tabla 3.3 Prueba U de Mann-Whitney sobre latencia unidireccional

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.4 Prueba U de Mann-Whitney sobre jitter unidireccional

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p = 0.000837$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 0.000418$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 0.9996$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.5 Prueba U de Mann-Whitney sobre ICPIF

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p = 7.866e-06$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 3.933e-06$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.6 Prueba U de Mann-Whitney sobre MOS

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p = 7.866e-06$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 3.933e-06$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.7 Prueba U de Mann-Whitney sobre RTT

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p = 0.000224$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 0.000112$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 0.9999$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es menor que la mediana del grupo 2 (VPN L3)</i>

Nota: La simulación no produjo pérdida de paquetes en ningún servicio MPLS VPN.

3.2.2. Métricas multicast

Tabla 3.8 Prueba U de Mann-Whitney sobre latencia unidireccional

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.9 Prueba U de Mann-Whitney sobre jitter unidireccional

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.10 Prueba U de Mann-Whitney sobre ICPIF

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.11 Prueba U de Mann-Whitney sobre MOS

Prueba	Estadístico	Descripción
<i>U de Mann-Whitney "Two-sided"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p < 2.2e-16$	<i>Se rechaza H_0, por lo tanto, la mediana del grupo 1 (VPN L2) es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 1$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es menor que la mediana del grupo 2 (VPN L3)</i>

Tabla 3.12 Prueba U de Mann-Whitney sobre RTT

<i>Prueba</i>	<i>Estadístico</i>	<i>Descripción</i>
<i>U de Mann-Whitney "Two-sided"</i>	$p = 0.4098$	<i>Se acepta H_0, por lo tanto, no existe una diferencia significativa entre las medianas de las métricas.</i>
<i>U de Mann-Whitney "Greater"</i>	$p = 0.2049$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es mayor que la mediana del grupo 2 (VPN L3)</i>
<i>U de Mann-Whitney "Less"</i>	$p = 0.7951$	<i>Se acepta H_0, por lo tanto, la mediana del grupo 1 (VPN L2) no es menor que la mediana del grupo 2 (VPN L3)</i>

Nota: La simulación no produjo pérdida de paquetes en ningún servicio MPLS VPN.

3.3. Intervalos de confianza (I.C.) de la media

Se utiliza Bootstrapping no paramétrico para estimar los intervalos de confianza de la media de cada conjunto de datos y compararlos entre sí. El nivel de significancia es 0.05 y el número de iteraciones es 10000. El algoritmo utilizado se muestra a continuación:

Algoritmo 2 Algoritmo para calcular los intervalos de confianza de la media

Requiere: $x_n, B, s(x_n)$

- 1: for i in 1:B
 - 2: Calcular x_n^* = re-muestreo con reemplazo de x_n
 - 3: Calcular $s(x_n^*)$
 - 4: Almacenar $v[i,1] = s(x_n) - s(x_n^*)$
 - 5: Calcular L = cuantil .025 de $v[,1]$
 - 6: Calcular U = cuantil .975 de $v[,1]$
 - 7: Calcular intervalos de confianza $L - s(x_n)$ y $s(x_n) - U$
-

La representación gráfica de los I.C. y la distribución de cada métrica se muestra en las secciones 3.3.1 y 3.3.2.

3.3.1. Métricas unicast

A. Latencia unidireccional

Descripción: La media del grupo 1 (VPNL2) es mayor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia es significativa.

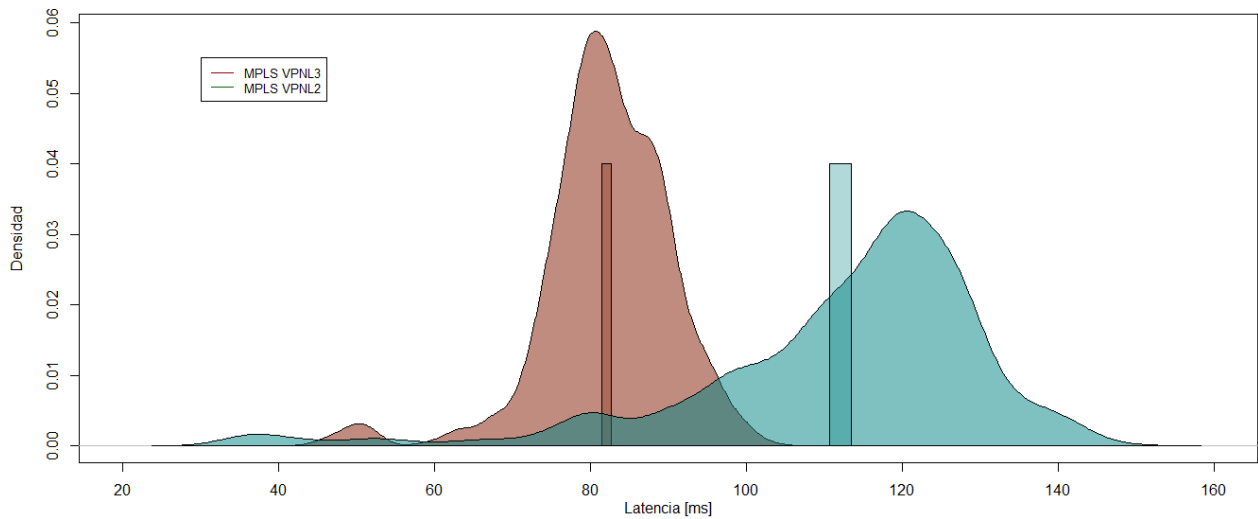


Fig 3.1 Distribución e I.C. de la latencia unidireccional en unicast

B. Jitter unidireccional

Descripción: La media del grupo 1 (VPNL2) es levemente mayor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia no es significativa.

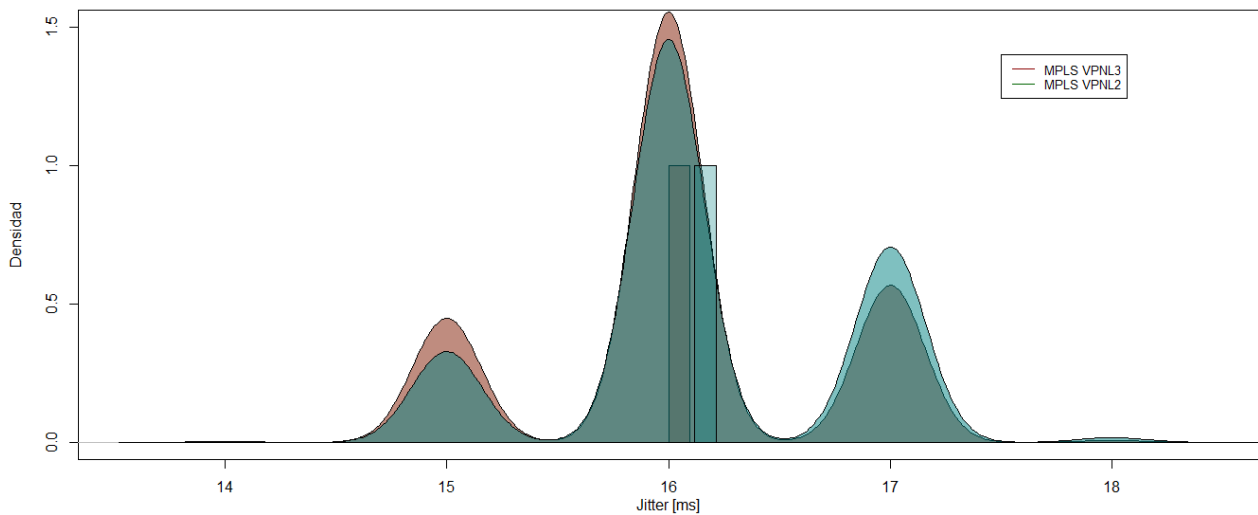


Fig 3.2 Distribución e I.C. del jitter unidireccional en unicast

C. ICPIF

Descripción: La media del grupo 1 (VPNL2) es levemente mayor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia no es significativa.

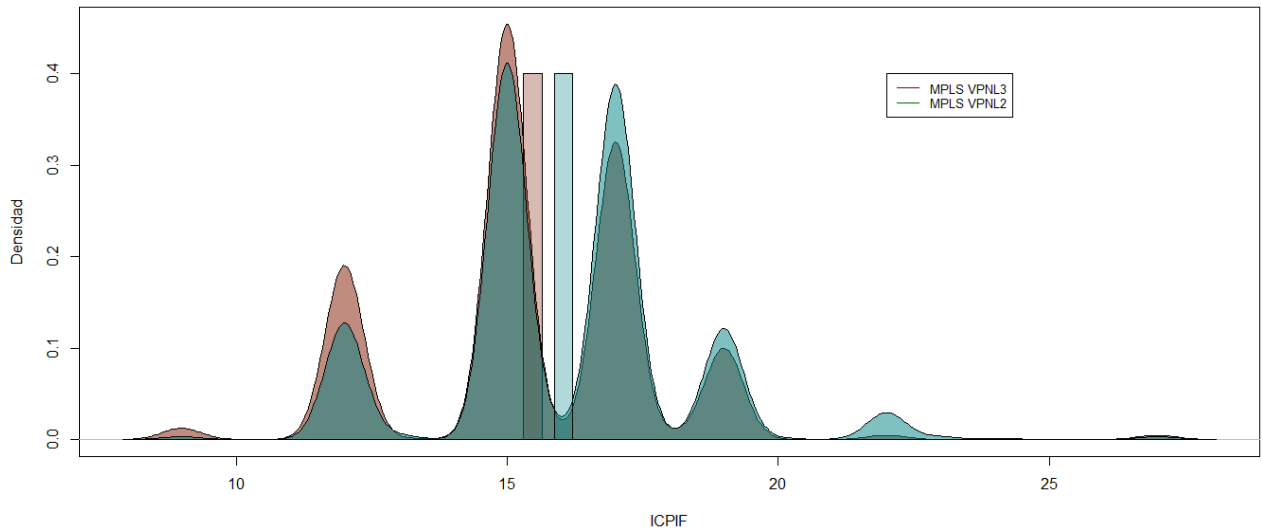


Fig 3.3 Distribución e I.C. del ICPIF en unicast

D. MOS

Descripción: La media del grupo 1 (VPNL2) es levemente menor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia no es significativa.

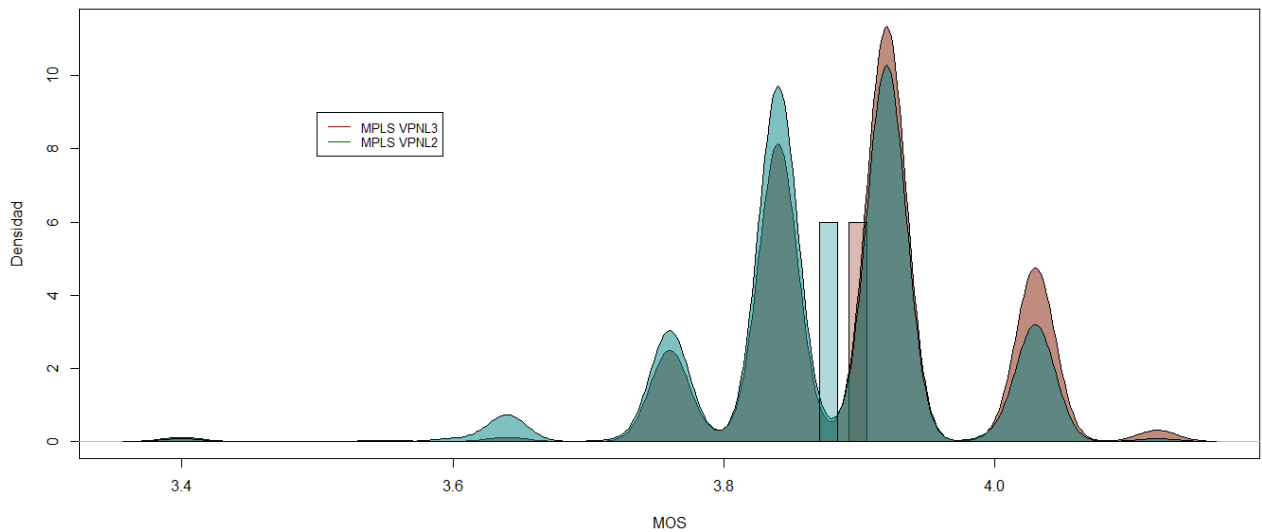


Fig 3.4 Distribución e I.C. del MOS en unicast

E. RTT

Descripción: Existe solapamiento de los C.I., por lo tanto, no existe diferencia estadísticamente significativa entre las medias de las métricas.

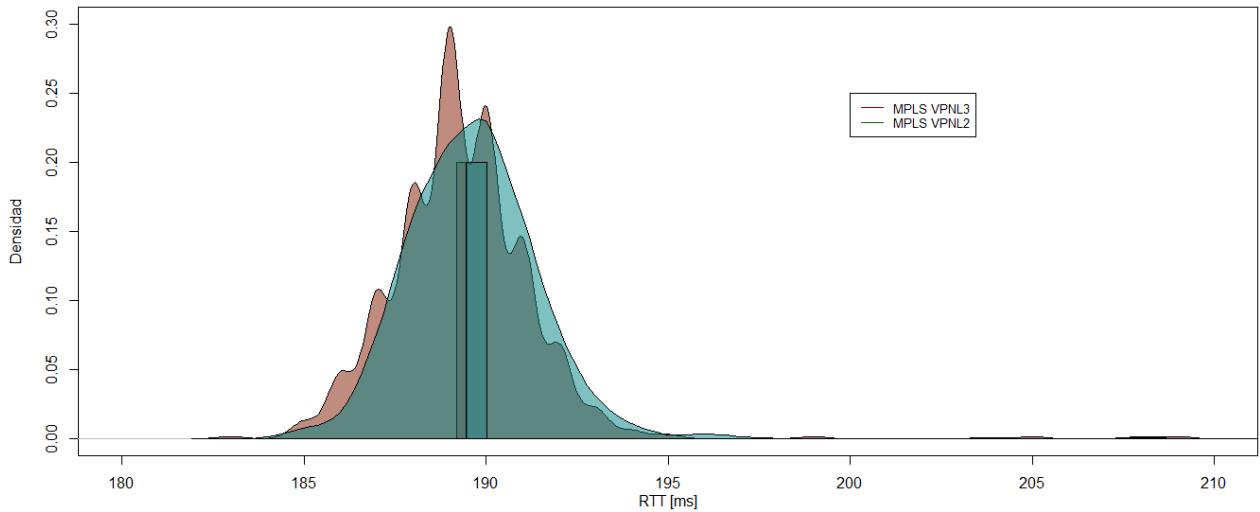


Fig 3.5 Distribución e I.C. del RTT en unicast

3.3.2. Métricas multicast

A. Latencia unidireccional

Descripción: La media del grupo 1 (VPNL2) es menor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia es poco significativa.

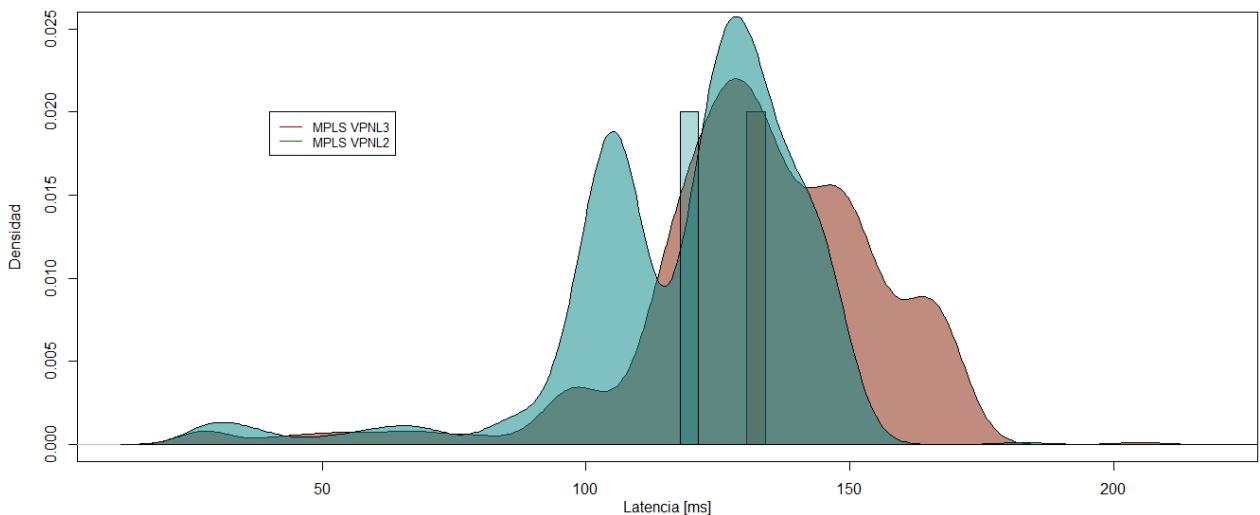


Fig 3.6 Distribución e I.C. de la latencia unidireccional en multicast

B. Jitter unidireccional

Descripción: La media del grupo 1 (VPNL2) es levemente mayor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia no es significativa.

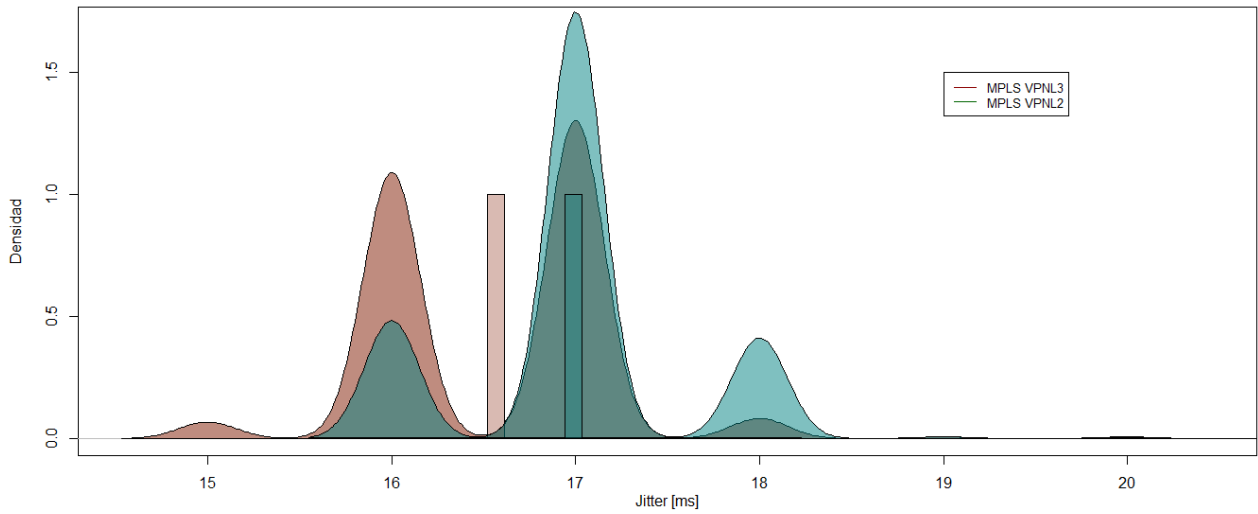


Fig 3.7 Distribución e I.C. del jitter unidireccional en multicast

C. ICPIF

Descripción: La media del grupo 1 (VPNL2) es levemente menor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia no es significativa.

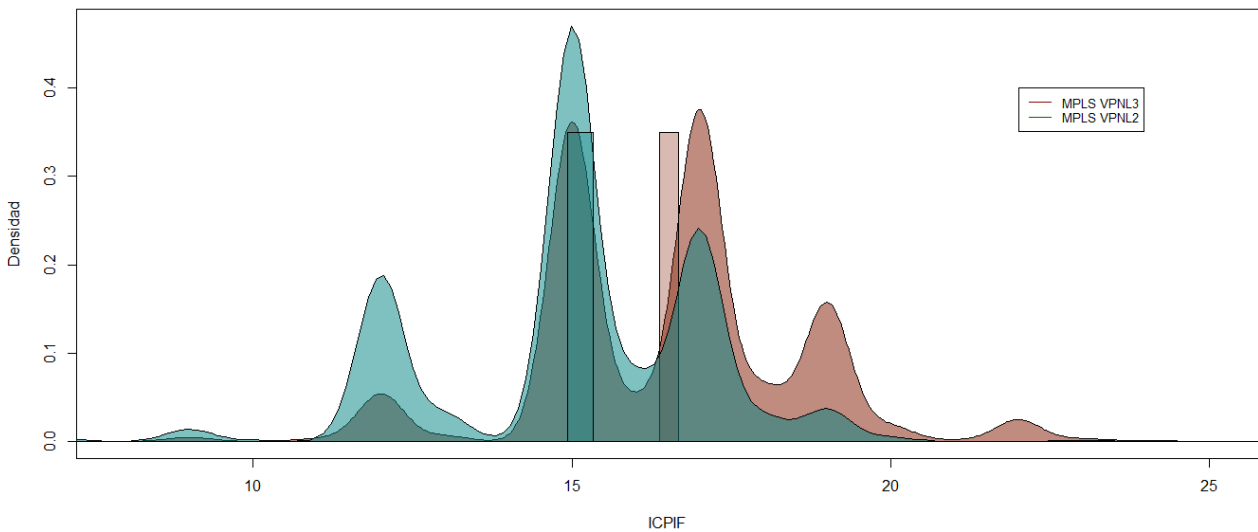


Fig 3.8 Distribución e I.C. del ICPIF en multicast

D. MOS

Descripción: La media del grupo 1 (VPNL2) es levemente mayor que la media del grupo 2 (VPNL3). En el contexto *networking*, esta diferencia no es significativa.

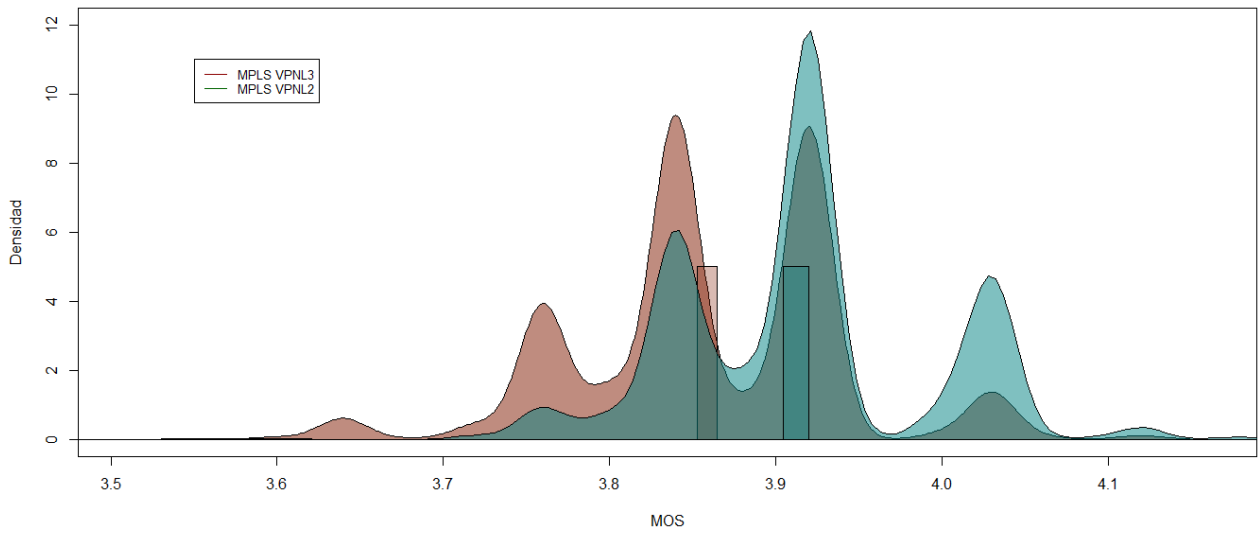


Fig 3.9 Distribución e I.C. del MOS en multicast

E. RTT

Descripción: Existe solapamiento de los C.I., por lo tanto, no existe diferencia estadísticamente significativa entre las medias de las métricas.

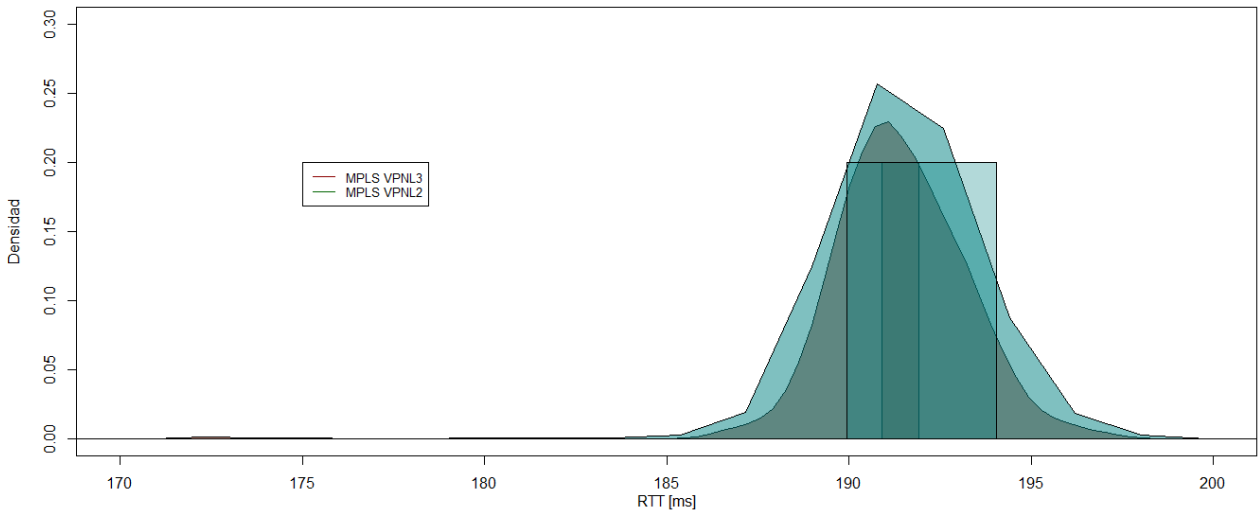


Fig 3.10 Distribución e I.C. del RTT en multicast

Capítulo 4

Conclusiones

En este capítulo se presentan las conclusiones obtenidas tras desplegar y simular los escenarios diseñados en la sección 2.1. Además, se analizan los resultados de los test estadísticos asociados a los objetivos e hipótesis planteadas. Para finalizar se exponen algunas apreciaciones sobre trabajos futuros.

4.1. Objetivos

Objetivo principal: *Cuantificar y comparar el desempeño de los servicios MPLS VPN L2 y L3.*

Este objetivo se cumple tras realizar una serie de procesos sistemáticos que involucran: diseño, implementación, simulación y análisis de datos. En el proceso de diseño se plantean escenarios experimentales que garantizan que el desempeño de los servicios MPLS VPN se ponga a prueba bajo las mismas condiciones. Estas condiciones comprenden nodos de borde y políticas de QoS comunes para los servicios que se contrastan. Además, los escenarios consideran la capacidad de los nodos emulados por Dynamips como restricción y el tráfico que atraviesa una red operativa como condición inicial. Este tráfico se modela con redes neuronales artificiales y para poder generarlo con IPERF, se utiliza BoxCox y Bootstrapping sobre el modelo para obtener estadísticos representativos. Los procesos de implementación y simulación se realizan sobre GNS3; este último comprende la ejecución simultánea y recurrente de IP SLA, kron, IPERF, Wireshark, NTP, TFTP y Dynamips. Finalmente, en el proceso de análisis de datos se utilizan test estadísticos sobre las métricas producidas por la simulación para realizar inferencias sobre el desempeño de los servicios MPLS VPN L2 y L3.

Los objetivos específicos tienen relación directa con los procesos realizados para cumplir el objetivo principal, por lo tanto, su cumplimiento es evidente.

4.2. Metodología

La metodología que se utiliza es aplicable a cualquier tipo de investigación que tiene por objeto el contraste de dos o más sistemas. Los procesos que la conforman están respaldados por consideraciones para evitar sesgos en los resultados y herramientas para realizar inferencias con validez estadística.

4.3. Hipótesis

Se desagrega la hipótesis planteada en función de sus métricas y se concluye:

1. Hipótesis: *El desempeño de los servicios MPLS VPN L2 en términos de delay es mejor comparado con los servicios MPLS VPN L3.* Con referencia en los test estadísticos, se puede afirmar que la latencia unidireccional de los servicios VPN L2 es considerablemente mayor con respecto a los servicios VPN L3 en unicast, mientras que en multicast, la latencia unidireccional de los servicios VPN L2 es levemente menor con respecto a los servicios VPN L3. Además, en unicast el RTT de los servicios VPN L2 es levemente mayor con

respecto a los servicios VPN L3, mientras que en multicast, esta métrica no presenta diferencias.

2. Hipótesis: *El desempeño de los servicios MPLS VPN L2 en términos de packet loss es mejor comparado con los servicios MPLS VPN L3.* Las simulaciones no produjeron pérdida de paquetes, por lo tanto, se puede afirmar que el desempeño de los servicios VPN L2 y L3 es similar en términos de esta métrica.
3. Hipótesis: *El desempeño de los servicios MPLS VPN L2 en términos de jitter es mejor comparado con los servicios MPLS VPN L3.* Con referencia en los test estadísticos, se puede afirmar que el jitter de los servicios VPN L2 es levemente mayor con respecto a los servicios VPN L3 en unicast y multicast.
4. Hipótesis: *El desempeño de los servicios MPLS VPN L2 en términos de MOS es mejor comparado con los servicios MPLS VPN L3.* Con referencia en los test estadísticos, se puede afirmar que el MOS de los servicios VPN L3 es levemente mejor con respecto a los servicios VPN L2 en unicast, mientras que en multicast, el MOS de los servicios VPN L2 es levemente mejor con respecto a los servicios VPN L3.
5. Hipótesis: *El desempeño de los servicios MPLS VPN L2 en términos de ICPIF es mejor comparado con los servicios MPLS VPN L3.* Con referencia en los test estadísticos, se puede afirmar que el ICPIF de los servicios VPN L2 es levemente mayor con respecto a los servicios VPN L3 en unicast, mientras que en multicast, el ICPIF de los servicios VPN L2 es levemente menor con respecto a los servicios VPN L3.
6. Es importante destacar que el desempeño de los servicios MPLS VPN L2 y L3 es bastante similar, por lo tanto, la elección de un determinado modelo debe realizarse en función de los costos iniciales (CAPEX por sus siglas en inglés, *Capital Expenditures*) y operación (OPEX por sus siglas en inglés, *Operating Expense*) y no de su desempeño.

4.4. Trabajos futuros

Desde la perspectiva del cliente, existe una tendencia por migrar sus servicios a soluciones de tipo SDN, sin embargo, los operadores que ofrecen esas soluciones utilizan internet como infraestructura para conectarlos con sus *data center*. Para servicios sensibles al retardo y pérdida de paquetes, migrar a SDN no es una solución adecuada ya que el proveedor no tiene control sobre esas métricas. Utilizando la misma metodología, se puede establecer un cluster de los servicios que pueden migrarse sin comprometer su performance.

Acrónimos

ACF Autocorrelation Function.

BGP Border Gateway Protocol.

COS Class Of Service.

EGP Exterior Gateway Protocol.

Eompls Ethernet over Mpls.

FIB Forwarding Information Base.

GNS3 Graphical Network Simulator 3.

ICPIF Impairment Calculated Planning Impairment Factor.

IGP Interior Gateway Protocol.

IP Internet Protocol.

LAN Local Área Network.

LDP Label Distribution Protocol.

LER Label Edge Router.

LFIB Label Forwarding Information Base.

LIB Label Information Base.

LSP Label Switched Path.

LSR Label Switching Router.

MAN Metropolitan Area Network.

MPLS Multiprotocol Label Switching.

MOS Mean Opinion Score.

NTP Network Time Protocol.

PACF Partial Autocorrelation Function.

QoS Quality Of Service.

RIB Routing Information Base.

RTT Round-Trip Time.

SLA Service Level Agreement.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

VM Virtual Machine.

VPN Virtual Private Network.

WAN Wide Area Network.

Bibliografía

- [1] U. Lakshman, L. Lobo, *MPLS Configuration on Cisco IOS Software: A complete configuration manual for MPLS, MPLS VPNs, MPLS TE, QoS, Any Transport over MPLS (AToM), and VPLS*, 2006.
- [2] P. Havrila, *Updated: L2 MPLS VPN introduction and H3C configuration examples (Martini and Kompella VLLs/VPLS)*, 2012. Disponible: <http://networkgeekstuff.com/networking/l2-mpls-vllsvpls-overview-including-martinikompella-mode-h3c-configuration-examples/>
- [3] N. Weinberg, And J. Till Johnson. *Como funciona MPLS*, 2018. Disponible: <http://www.networkworld.es/telecomunicaciones/como-funciona-mpls>
- [4] J. Won-Ki Hong, *Autonomic Traffic Monitoring and Analysis*, 2012. Disponible: https://austin.postech.ac.kr/guni/index.php/EECE702D:_Autonomic_Traffic_Monitoring_and_Analysis
- [5] N.D. Lewis, *Neural Networks For Time Series Forecasting with R*, 2017.
- [6] M. Bernacki, P. Włodarczyk, *Principles of training multi-layer neural network using backpropagation*, 2005. Disponible: http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html
- [7] J. Gross ,U. Ligges, *Package 'nortest'*, 2015. Disponible: <https://cran.r-project.org/web/packages/nortest/nortest.pdf>
- [8] D. Kuonen, *An Introduction to Bootstrap Methods and their Application*, 2018. Disponible: https://www.ethz.ch/content/dam/ethz/special-interest/math/statistics/sfs/Education/Advanced%20Studies%20in%20Applied%20Statistics/course-material-1719/Nonparametric%20Methods/lecture_2up.pdf
- [9] J. Orloff and J. Bloom, *Bootstrap confidence intervals*, 2014. Disponible: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading24.pdf
- [10] S. Millard, *Boxcox Power Transformation*, 2018. Disponible: <https://www.rdocumentation.org/packages/EnvStats/versions/2.3.0/topics/boxcox>
- [11] S. M. Karadimitriou and E. Marshall, 2018. *Mann-Whitney U test in R*, Disponible: https://www.sheffield.ac.uk/polopoly_fs/1.714563!/file/stcp-karadimitriou-MannWhitR.pdf
- [12] The R Foundation, *What is R?*, 2018. Disponible: <https://www.r-project.org/about.html>
- [13] Community Cisco, *BGP, PAK-Priority, and QoS*, 2009. Disponible: <https://community.cisco.com/t5/routing/bgp-pak-priority-and-qos/td-p/1397656>

Anexos

A. Códigos R

1. Modelación del tráfico con redes neuronales artificiales

```
rm(list = setdiff(ls(), lsf.str()))

library(caret)

library(forecast)

library(devtools)

source_url('https://gist.githubusercontent.com/fawda123/7471137/raw/466c1474d0a505ff044412703516c34f1a4684a5/nnet_plot_update.r')

require(rnn)

require(quantmod)

require(reshape)

lagf <- function(datos,lag) {
  for (i in 1:lag) {
    r = Lag(datos[,1], k=i)
    dl = cbind(datos,r)
    datos = dl
  }
  return(datos)
}

normalize <- function (m){
  m <- (m - min(m)) / (max(m) - min(m))
  return(m)
}

denormalize <- function(x,minval,maxval) {
  return(x*(maxval-minval) + minval)
}

dataU <- read.table("C:\\Users\\jorge\\OneDrive\\MIRC\\Tesis - Modelamiento\\Entel - VTR - Up.dat",h=T)
dataD <- read.table("C:\\Users\\jorge\\OneDrive\\MIRC\\Tesis - Modelamiento\\Entel - VTR - Dw.dat",h=T)
```

```
## NORMALIZAMOS LOS DATOS ##
```

```
xU = as.matrix(c(max(dataU$Kbps),min(dataU$Kbps)))
```

```
xD = as.matrix(c(max(dataD$Kbps),min(dataD$Kbps)))
```

```
dataU = normalize(dataU)
```

```
dataD = normalize(dataD)
```

```
plot(dataD$Kbps ~ dataD$Mes,type = "l",col="red",xlab="Mes",ylab="Ocupación [Kbps]")
```

```
lines(dataU$Kbps ~ dataU$Mes,type = "l",col="darkblue")
```

```
grid (NULL,NULL, lty = 6, col = "lightgray")
```

```
## CREAMOS CLUSTER DE ENTRENAMIENTO Y VALIDACION ##
```

```
lag = 5
```

```
dU = lagf(as.matrix(dataU$Kbps),lag)
```

```
dD = lagf(as.matrix(dataD$Kbps),lag)
```

```
head(dU)
```

```
head(dD)
```

```
dU <- dU[-(1:lag),]
```

```
dD <- dD[-(1:lag),]
```

```
dataU <- data.frame(dU)
```

```
dataD <- data.frame(dD)
```

```
colnames(dataU)=c("datos", "Lag.1", "Lag.2", "Lag.3", "Lag.4", "Lag.5")
```

```
colnames(dataD)=c("datos", "Lag.1", "Lag.2")
```

```
hist(dataU$datos)
```

```
hist(dataD$datos)
```

```
u <- round(0.8*nrow(dataU))
```

```
dEU <- dataU[1:u,]
```

```
dVU <- dataU[(u+1):nrow(dataU),]
```

```

u <- round(0.8*nrow(dataD))
dED <- dataD[1:u,]
dVD <- dataD[(u+1):nrow(dataD),]

      ## CONSTRUCCION MODELO CON REDES NEURONALES ##

      ## TRÁFICO UPLINK ##

set.seed(10)
fitControl <- trainControl(## 10 fold Cross Validation
  method = "repeatedcv",
  number = 10,
  ## repeated ten times
  repeats = 50
  )

nnmU <- train(datos ~., data = dEU,
  method = "nnet",
  trControl = fitControl,
  trace = F, linout = 1
)

wts.in = nnmU$finalModel$wts
struct = nnmU$finalModel$struct
plot.nnet(wts.in,struct=struct)
summary(nnmU$finalModel)

#saveRDS(nnmU, "FinalModelU.rds")

nnpU <- na.omit(predict(nnmU,dVU,type="raw"))
nnpU <- denormalize(nnpU,254555.5,704555.5)
dVU$datos <- denormalize(dVU$datos,254555.5,704555.5)

plot(dVU$datos,type="l",col="black")

```



```

lines(nnpU,lty = 2,col="red")
grid (NULL,NULL, lty = 6, col = "lightgray")

nnmapeU <- mean(abs((nnpU - dVU$datos))/dVU$datos)
nnmapeU*100

nnrmseU <- sqrt(mean((nnpU - dVU$datos)^2))
nnrmseU

res_v = dVU$datos-nnpU
mean(res_v)
sd(res_v)

                                ## DOWNLINK ##

set.seed(10)
fitControl <- trainControl(method = "repeatedcv",
                            number = 10,
                            repeats = 100)

nnmD <- train(datos ~., data = dED,
              method = "nnet",
              trControl = fitControl,
              trace = F, linout = 1 # PARA QUE FUNCIONE COMO REGRESION Y NO CLASIFICACION
)
saveRDS(nnmD, "FinalModelD2018.rds")

wts.in = nnmD$finalModel$wts
struct = nnmD$finalModel$n
plot.nnet(wts.in,struct=struct)

nnpD <- na.omit(predict(nnmD,dVD,type="raw"))
nnpD <- denormalize(nnpD,109100.9,709100.9)
dVD$datos <- denormalize(dVD$datos,109100.9,709100.9)

```

```
plot(dVD$datos,type="l",col="black")
lines(nnpD,lty=2,col="red")
grid(NULL,NULL, lty = 6, col = "lightgray")

nnmaped <- mean(abs((nnpD -dVD$datos))/dVD$datos)
nnmaped*100

nnrmseD <- sqrt(mean((nnpD - dVD$datos)^2))
nnrmseD

res_v = dVD$datos-nnpD
mean(res_v)
sd(res_v)
```

2. Transformación con Box-Cox y estimación de la media poblacional

```
rm(list = setdiff(ls(), lsf.str()))
library(nortest)
library(boot)

media <- function(datos, indices) {
  d <- mean(datos[indices])
  return(d - md)
}

denormalize <- function(x,minval,maxval) {
  return(x*(maxval-minval) + minval)
}

mfU <- readRDS("FinalModelU.rds")
vnn <- denormalize(mfU$finalModel$fitted.values[,1],254555.5,704555.5)

B = 10^4
r1 = numeric(B)
set.seed(10)
mm <- boot(data=vnn, statistic=media, R=B)$t
#saveRDS(mm,"MeanMatrix.rds")

mm <- readRDS("MeanMatrix.rds")
x <- mm[,1]

lambda = 0.60905
T = as.matrix((x^lambda - 1)/lambda)
ad.test(T)
mean(T)
```

3. Test no paramétricos y estimación de los I.C. de la media

```
rm(list = setdiff(ls(), lsf.str()))
library(boot)
library(nortest)

media <- function(datos, indices) {
  d <- mean(datos[indices])
  return(d - md)
}

ud2 <- read.table("C:\\Users\\jorge\\OneDrive\\MIRC\\Tesis Estadisticas\\uL2.txt",h=T)
ud3 <- read.table("C:\\Users\\jorge\\OneDrive\\MIRC\\Tesis Estadisticas\\uL3.txt",h=T)

##### PRUEBAS DE NORMALIDAD #####

nt = as.matrix(ud2)
r1 = matrix(NA,1,ncol(nt))
for (i in 1:ncol(nt)){
  tryCatch({
    r1[1,i]=ad.test(ud2[,i])$p.value
  }, error=function(e){ })
}

##### IC con libreria BOOT #####
##### LAYER 2 #####

B = 10^4
dl2 = as.matrix(ud2)
r2 = matrix(NA,B,ncol(dl2))

for (i in 1:ncol(dl2)){
  md = mean(dl2[,i])
  set.seed(10)
  r2[,i] <- boot(data=dl2[,i], statistic=media, R=B)$t
}
```

```

icl2 = matrix(NA,ncol(r2),2)          ## | L | U |
for (i in 1:ncol(r2)){
  md = mean(dl2[,i])
  icl2[i,1] = md - quantile(r2[,i], .025)
  icl2[i,2] = md - quantile(r2[,i], .975)
}
icl2  # Los Lod Js Jd ICPIF MOS Ps Pd Pt Pp RTT

```

LAYER 3

```

B = 10^4
dl3 = as.matrix(ud3)
r3 = matrix(NA,B,ncol(dl3))

for (i in 1:ncol(dl3)){
  md = mean(dl3[,i])
  set.seed(10)
  r3[,i] <- boot(data=dl3[,i], statistic=media, R=B)$t
}

```

```

icl3 = matrix(NA,ncol(r3),2)        ## | L | U |
for (i in 1:ncol(r3)){
  md = mean(dl3[,i])
  icl3[i,1] = md - quantile(r3[,i], .025)
  icl3[i,2] = md - quantile(r3[,i], .975)
}
icl3  # Los Lod Js Jd ICPIF MOS Ps Pd Pt Pp RTT

```

GRÁFICOS DE C.I.

p = 11

a = 0.2

```

plot(density(dl3[,p]), main="Distribución",ylab="Densidad",
      xlab = "RTT [ms]\n",xlim=c(180,210),ylim=c(0,0.3))
polygon(density(dl3[,p]), col=rgb(0.5,0.1,0,0.5))

```

```
rect(icl3[p,2], 0, icl3[p,1], a, col=rgb(0.5,0.1,0,0.3))

#plot(density(dl2[p]), main="Distribución",ylab="Densidad")
polygon(density(dl2[p]), col=rgb(0,0.51,0.51,0.5))
rect(icl2[p,2], 0, icl2[p,1], a, col=rgb(0,0.51,0.51,0.3))

legend(200,0.25, legend=c("MPLS VPNL3", "MPLS VPNL2"),
       col=c("darkred", "darkgreen"),lty=1:1 ,cex=0.8)
```