



Universidad Nacional Pedro Ruiz Gallo
Facultad de Ciencias Físicas y Matemáticas
Escuela Profesional de Ingeniería Electrónica



Tesis

**Desempeño de VoIP en redes de banda ancha en localidades rurales del
distrito de Huacaspata, Pataz, región La Libertad**

para optar el título profesional de Ingeniero Electrónico

presentado por:

Bach. Carrillo Zeña Jans Robinson

Asesor:

Ing. Segundo Francisco Segura Altamirano.

Lambayeque - Perú

10 de diciembre de 2025

Universidad Nacional Pedro Ruiz Gallo
Facultad de Ciencias Físicas y Matemáticas
Escuela Profesional de Ingeniería Electrónica

**Desempeño de VoIP en redes de banda ancha en
localidades rurales del distrito de Huacaspata, Pataz,
región La Libertad**

Autor:

Bach. Carrillo Zeña Jans Robinson



Asesor:

Ing. Segundo Francisco Segura Altamirano.



LAMBAYEQUE - PERÚ

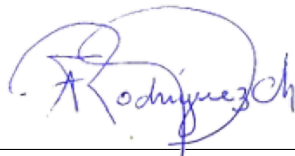
10 de diciembre de 2025

**Universidad Nacional Pedro Ruiz Gallo
Facultad de Ciencias Físicas y Matemáticas
Escuela Profesional de Ingeniería Electrónica**

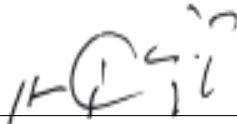
**Desempeño de VoIP en Redes de Banda Ancha en Localidades Rurales
del Distrito de Huacaspata, Pataz, Región La Libertad**

Tesis para optar el título profesional de Ingeniero Electrónico

Sustentado y aprobado ante los siguientes miembros del jurado:



M.Sc. Ing. Frank Richard Rodriguez Chirinos.
Presidente



MS.c. Ing. Hugo Javier Chiclayo Padilla.
Secretario



Mg. Ing. Martín Augusto Nombera Lossio.
Vocal

Lambayeque - Perú

10 de diciembre de 2025



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DECANATO

Ciudad Universitaria – Lambayeque

LICENCIADA - RESOLUCIÓN DEL CONSEJO DIRECTIVO N° 015 -2023-SUNEDU / CD



0200



ACTA DE SUSTENTACIÓN N° 83.-2025.-D/FACFyM

Siendo las 7:30 am del día 10 de diciembre del 2025, se reunieron los miembros del jurado evaluador de la Tesis titulada:

Desempeño de VoIP en Redes de Banda Ancha en localidades Rurales del Distrito de Huacaspata, Patate, Región La Libertad

Designados por Resolución N° 1093-2023 virtual D/FACFyM de fecha 29 noviembre 2023

Con la finalidad de evaluar y calificar la sustentación de la tesis antes mencionada, conformada por los siguientes docentes:

<u>Dr. Ing. Frank Richard Rodríguez Chirinos</u>	Presidente
<u>Mg. Ing. Hugo Javier Chiclayo Padilla</u>	Secretario
<u>Mg. Ing. Martín Augusto Nomberra Lasso</u>	Vocal

La tesis fue asesorada por (el) (la) M.Sc. Ing. Segundo Francisco Segura Altamirano nombrado por Resolución N° 1093-2023-Virtual D/FACFyM de fecha 29 noviembre 2023

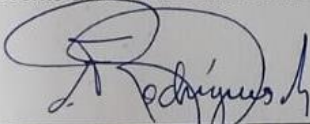
El Acto de Sustentación fue autorizado por Resolución N° 118-2025 D/FACFyM de fecha 25 de noviembre de 2025

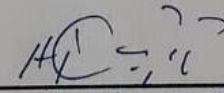
La Tesis fue presentada y sustentada por (el) (los) Bachiller (es): Carrillo Zeña Jans Robinson y tuvo una duración de 38 minutos.

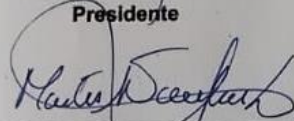
Después de la sustentación, y absueltas las preguntas y observaciones de los miembros del jurado se procedió a la calificación respectiva, otorgándole el Calificativo de Quince (15) en la escala vigesimal, mención (Regular).


Por lo que queda(n) apto(s) para obtener el Título Profesional de Ingeniero Electrónico de acuerdo con la Ley Universitaria 30220 y la normatividad vigente de la Facultad de Ciencias Físicas y Matemáticas y la Universidad Nacional Pedro Ruiz Gallo.

Siendo las 8:45 am se dio por concluido el presente acto académico, dándose conformidad al presente acto con la firma de los miembros del jurado.


DR. ING. FRANK RICHARD RODRIGUEZ CHIRINOS
 Presidente


Mg. Ing. Hugo Javier Chiclayo Padilla
 Secretario


Mg. Ing. Martín Augusto Nomberra Lasso
 Vocal


MSc. Ing. Segundo Francisco Segura Altamirano
 Asesor

ORIGINALITY REPORT

5%

SIMILARITY INDEX

5%

INTERNET SOURCES

2%

PUBLICATIONS

2%

STUDENT PAPERS

PRIMARY SOURCES

1	repositorio.unprg.edu.pe Internet Source	3%
2	repositorio.uct.edu.pe Internet Source	<1%
3	catalonica.bnc.cat Internet Source	<1%
4	qdoc.tips Internet Source	<1%
5	repositorio.ug.edu.ec Internet Source	<1%
6	kupdf.net Internet Source	<1%
7	Submitted to Instituto Tecnológico de Costa Rica Student Paper	<1%
8	sedici.unlp.edu.ar Internet Source	<1%
9	repositorio.espe.edu.ec Internet Source	<1%

10	docplayer.es Internet Source	<1 %
11	ri.ues.edu.sv Internet Source	<1 %
12	Hernán Óscar Cortez Gutiérrez, Milton Milcíades Cortez Gutiérrez. "Regresión logística en la clasificación y riesgos para desempeño profesional", TECHNO REVIEW. International Technology, Science and Society Review /Revista Internacional de Tecnología, Ciencia y Sociedad, 2023 Publication	<1 %
13	www.dropbox.com Internet Source	<1 %

Exclude quotes Off

Exclude matches < 15 words

Exclude bibliography On



Segundo Francisco Segura Altamirano

DNI: 09922587

Tesis Final

GRADEMARK REPORT

FINAL GRADE

/0



GENERAL COMMENTS

Segundo Francisco Segura Altamirano.
DNI: 09922587

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

Segundo Francisco Segura Altamirano.
DNI: 09922587

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66



Segundo Francisco Segura Altamirano.
DNI: 09922587

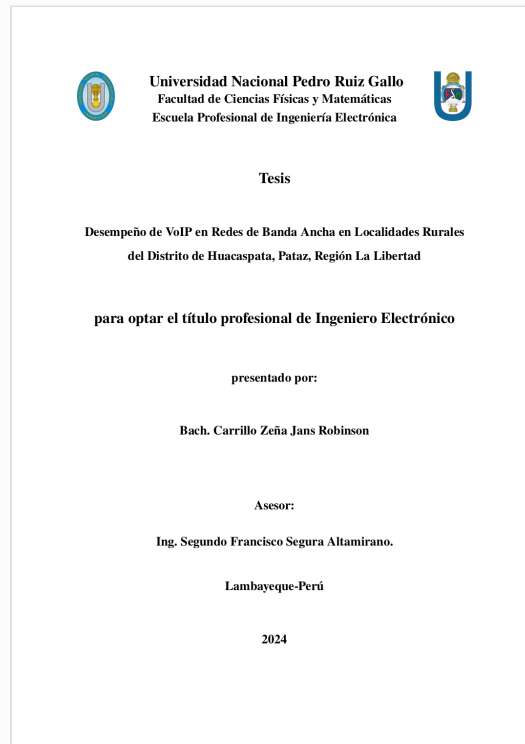


Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Jans Robinson Carrillo Zeña
Assignment title: Quick Submit
Submission title: Tesis Final
File name: mainCarrillo.pdf
File size: 12.82M
Page count: 66
Word count: 14,191
Character count: 79,129
Submission date: 11-Dec-2024 12:45PM (UTC-0500)
Submission ID: 2549241695




Segundo Francisco Segura Altamirano
DNI: 09922587

Agradecimientos

El presente trabajo de investigación se logró concretizar, gracias a la orientación de mi asesor MSc. Ing. Segundo Francisco Segura Altamirano, quien contribuyó de forma esencial con sus conocimientos y experiencia, guiándome de manera clara y precisa, para desarrollar de manera exitosa la presente Tesis, que constituye una línea base para futuras investigaciones.

A mi Alma Mater, por acogerme en sus aulas y brindarme los conocimientos esenciales, con el cual me ha permitido afrontar de manera exitosa los retos al que todo profesional es sometido.

A Jehová Dios y su hijo Jesucristo por darme la vida, salud y su bondad inmerecida.

Gracias por todo.

Bach. Carrillo Zeña Jans Robinson

Bachiller Ingeniería Electrónica

Lambayeque 2025

Dedicatoria

A mis padres Don Teodoro Carrillo Carlos y Doña América Zeña Piscoya, que en todo momento dedicaron sus esfuerzos para brindarme la mejor herencia que todo padre anhela para su hijo, por impulsarme a lograr mis metas y poder consolidarme como profesional y persona de bien para la sociedad.

A mis hermanos Jhonny Carrillo y Edwduar Carrillo, por su apoyo incondicional e impulsarme a ser mejor cada día.

Gracias por todo.

Bach. Carrillo Zeña Jans Robinson

Bachiller Ingeniería Electrónica

Lambayeque 2025

Resumen

Este estudio aborda la problemática del desempeño de servicios VoIP en redes de banda ancha en zonas rurales del distrito de Huacaspata, Pataz, donde la elección adecuada del códec es crucial para garantizar un servicio de calidad con recursos limitados. El objetivo fue evaluar el impacto de diferentes códecs en la asignación de recursos de la red inalámbrica de banda ancha, utilizando una metodología basada en simulaciones con NS3 para analizar métricas clave como latencia, jitter y pérdida de paquetes. Los resultados revelaron que el G.729 emerge como la opción más eficiente, requiriendo solo 8 Kbps mientras mantiene una calidad aceptable (MOS 3.92), permitiendo hasta 34 llamadas simultáneas por Mbps, en contraste con las 11 llamadas del G.711. El análisis de latencia mostró que el G.729 presenta los menores retardos (0.05-0.085 ms), mientras que el jitter se mantuvo consistente entre todos los códecs ($0 - 0,610^{-13}$ segundos), sugiriendo una mayor influencia de la infraestructura de red que del códec utilizado. Se concluye que la elección del códec impacta significativamente en la eficiencia de la red, siendo el G.729 particularmente adecuado para entornos rurales por su óptimo balance entre calidad de servicio y consumo de recursos.

Palabras clave: VoIP Rural, Códecs de Voz, Redes de Banda Ancha, Calidad de Servicio, Telecomunicaciones Rurales.

Abstract

This study addresses the issue of VoIP service performance in broadband networks in rural areas of the Huacaspata district, Pataz, where the proper choice of codec is crucial to ensure quality service with limited resources. The objective was to evaluate the impact of different codecs on the allocation of resources in the broadband wireless network, using a simulation-based methodology with NS3 to analyze key metrics such as latency, jitter, and packet loss. The results revealed that G.729 emerges as the most efficient option, requiring only 8 Kbps while maintaining acceptable quality (MOS 3.92), allowing up to 34 simultaneous calls per Mbps, in contrast to the 11 calls of G.711. Latency analysis showed that G.729 exhibits the lowest delays (0.05-0.085 ms), while jitter remained consistent across all codecs ($0 - 0,610^{-13}$ segundos), suggesting a greater influence of network infrastructure than the codec used. It is concluded that the choice of codec significantly impacts network efficiency, with G.729 being particularly suitable for rural environments due to its optimal balance between quality of service and resource consumption.

Key words: VoIP rural, Voice codecs, Broadband networks, Quality of Service, Rural telecommunications.

Índice general

Reconocimientos	v
Dedicatoria	vi
Resumen	vii
Abstract	viii
Índice de figuras	xii
Índice de Tablas	xiii
Introducción	xiv
1. Capítulo I	1
1.1. Situación problemática	1
1.2. Formulación del problema de investigación	2
1.3. Hipótesis	2
1.4. Objetivos de la investigación	2
1.4.1. Objetivo General	2
1.4.2. Objetivos Específicos	3
2. Capítulo II: Marco Teórico	4
2.1. Antecedentes	4
2.2. Bases Teóricas	5
2.2.1. Análisis de Población del distrito de Huacaspata	5
2.2.1.1. Localidades beneficiadas	8
2.2.2. Modelos de tráfico VoIP fijo y móvil	9
2.2.2.1. Codecs de Voip	9
2.2.2.2. Modelos de Poisson y Markov	11
2.2.2.3. Modelo de Fuente On/Off	11

2.2.2.4.	Multiplexación de Fuentes On/Off	12
2.2.2.5.	Modelos de Pareto	12
2.2.3.	Software de Planificación y Simuladores	13
2.2.3.1.	El Presupuesto de Enlace	15
2.2.3.2.	Factores Adicionales en la Planificación	16
2.2.3.3.	Proceso Integral de Planificación	16
2.2.3.4.	Radio Mobile: La Alternativa de Código Abierto	16
2.2.3.5.	RadioPlanner 3.0	17
2.2.3.6.	LinkPlanner	18
2.2.3.7.	Fundamentos del Simulador NS3	19
2.2.3.8.	Simulación de VoIP en NS3	19
2.2.3.9.	Generación de Tráfico y Análisis	19
2.2.3.10.	Métricas y Evaluación de Rendimiento	20
3.	Capítulo III: Marco Metodológico	21
3.1.	Introducción	21
3.1.1.	Tipo de Investigación	21
3.1.2.	Diseño de Investigación	21
3.1.2.1.	No experimental	21
3.1.2.2.	Transversal	22
3.1.2.3.	Comparativo	22
3.1.2.4.	Justificación del Diseño	22
3.2.	Contraste de Hipótesis	22
3.3.	Población y Muestra	23
3.4.	Técnicas e Instrumentos de Recolección de Datos	23
3.4.1.	Técnicas	24
3.4.2.	Instrumentos	24
3.5.	Análisis y Procesamiento de Datos	24

4. Capítulo IV: Resultados	26
4.1. Estimación de Demanda	26
4.2. Red de Acceso Hucaspata	27
4.3. Simulador con Diferentes Codec	28
4.3.1. Simulación de Red de Acceso de Localidad de Huacaspata - Tra-	
fico VoIP	31
4.3.1.1. Trafico Entrante: Generado y en la Red	31
4.3.1.2. Trafico Entrante: Generado y en la Red	34
4.3.1.3. Retardo y Jitter	37
5. Conclusiones	41
APPENDICES	43
Carpeta compartida y programa usado	44

Índice de figuras

2.1. Análisis de Población y Tiempo de viaje del distrito de Huacaspata	7
2.2. Gráfico de Dispersión de Población y Tiempo de viaje	8
3.1. Proceso Metodológico de Investigación	23
4.1. Comparación de Ancho de Banda - Promedio Localidades	27
4.2. Reprote de Capacidad de Celdas - Localidad de Huacaspata	28
4.3. Red de Acceso Localidad de Huacaspata	29
4.4. Bloques de Simulador	31
4.5. Trafico de 28 Llamadas y Diferentes Codecs	33
4.6. Trafico de 24 Llamadas y Diferentes Codecs	35
4.7. Retardo y Jitter de Localidad de Huacaspata	37
4.8. Retardo Promedio por Localidad y codec	40
4.9. Jitter Promedio por Localidad y codec	40

Índice de Tablas

2.1. Análisis descriptivo: Localidades de Huacaspata	6
2.2. Localidades Beneficiadas usando los criterios de ruralidad de Pronatel . .	8
2.3. Demanda Estimada Total: Considerando codec G711	10
2.4. Codecs VoIP usados en Simulación	11
4.1. Capacidad Estimada de Trafico VoIP con Diferentes Codecs	26

Introducción

En el contexto actual de la sociedad digital, el acceso a servicios de telecomunicaciones de calidad se ha convertido en un elemento fundamental para el desarrollo socioeconómico, especialmente en áreas rurales. Este estudio se centra en el análisis del desempeño de servicios VoIP en redes de banda ancha en el distrito de Huacaspata, región La Libertad, Perú, donde la optimización de recursos es crucial para garantizar una conectividad efectiva.

Los datos de la ONU sobre el Estado de la Banda Ancha 2024 revelan una brecha digital persistente: mientras que el 67% de la población mundial tiene acceso a Internet, existe un significativo 38% que, aun teniendo cobertura, no utiliza estos servicios, y un 5% que carece completamente de acceso. Esta situación es particularmente crítica en zonas rurales, donde la pandemia de COVID-19 ha expuesto dramáticamente las desigualdades en el acceso a la conectividad digital.

El estudio se estructura en cuatro capítulos fundamentales. El Capítulo I presenta el marco contextual, definiendo la problemática de la conectividad rural y estableciendo los objetivos específicos de la investigación. El Capítulo II desarrolla el marco teórico, explorando los diferentes códecs VoIP, sus características técnicas y su impacto en la calidad del servicio, además de analizar las herramientas de planificación y simulación disponibles.

El Capítulo III detalla la metodología empleada, basada en un enfoque no experimental y transversal, utilizando simulaciones con NS3 para evaluar el comportamiento de diferentes códecs en condiciones reales. Se analizan métricas cruciales como latencia, jitter y pérdida de paquetes en una muestra de 13 localidades rurales seleccionadas según criterios específicos de ruralidad. El Capítulo IV presenta los resultados del estudio, revelando hallazgos significativos sobre el desempeño de los códecs. Destaca particularmente el G.729 por su eficiencia en el uso de recursos, manteniendo una calidad de servicio aceptable, lo que lo posiciona como una opción especialmente adecuada para entornos rurales con limitaciones de infraestructura.

Esta investigación contribuye significativamente al campo de las telecomunicaciones rurales, proporcionando evidencia empírica sobre la optimización de servicios VoIP en contextos de recursos limitados. Los resultados tienen implicaciones prácticas para la planificación y despliegue de redes de banda ancha en áreas rurales, donde la eficiencia en el uso de recursos es crucial para garantizar la sostenibilidad y calidad del servicio.

Finalmente, el estudio se enmarca en el contexto más amplio del Proyecto Nacional de Banda Ancha del Perú, que busca extender la conectividad digital a todo el territorio nacional. Los hallazgos y recomendaciones presentados aquí pueden servir como referencia para futuras implementaciones en contextos rurales similares, contribuyendo así al objetivo más amplio de reducir la brecha digital en el país.

Capítulo I

1.1. Situación problemática

Conectarse a Internet es vital para desarrollarse y salir adelante en nuestra sociedad actual. Según el documento de la ONU: El Estado de la Banda Ancha 2024, publicado en agosto de este año, las cifras del 2023 muestran que aunque más de la mitad del mundo (67%) puede conectarse, con unos 5.4 mil millones de personas en línea, todavía hay lugares donde la gente no tiene acceso, sobre todo en el campo y en comunidades que necesitan más apoyo. Me llama la atención que aunque hay buena señal de Internet móvil en muchos sitios, hay un 38% de gente que no lo usa aunque podría, y un 5% que ni siquiera tiene la posibilidad de conectarse.

Frans nos dice en su investigación que "la pandemia del Covid-19 dejó ver claramente qué tan desigual es el acceso a información por Internet de banda ancha, especialmente en zonas alejadas del campo"(2022). Los gobiernos han tratado de ayudar poniendo centros públicos con computadoras e Internet, pero estas iniciativas no siempre funcionan bien por problemas como la corrupción y porque algunas comunidades quedan fuera del proyecto.

En cuanto a las nuevas tecnologías de comunicación, el 5G está haciendo que todo se vuelva digital más rápido. Se espera que las suscripciones aumenten de 1.6 mil millones en 2023 a 3.7 mil millones para 2027. Pero como menciona Maslen, "incluso en países ricos como Estados Unidos, hay millones de personas que todavía no tienen un Internet que funcione bien"(2021).

Cuando hablamos de servicios importantes, la telefonía por Internet o VoIP es clave en esta red digital. Como explica Iskandar, hay retos grandes para manejar bien estos servicios, y propone usar contestadores automáticos con inteligencia artificial como solución (2024). Para emergencias, según señala Ghassan, las redes MANET son muy útiles porque permiten hacer llamadas VoIP sin necesitar antenas fijas ni cables (2021).

En Perú, el Proyecto Nacional de Banda Ancha está trabajando de forma organizada

para llevar Internet a todo el país. Han instalado cables de fibra óptica que conectan las ciudades principales y llegan hasta los distritos más pequeños.

En Huacaspata, un distrito de Pataz en La Libertad, podemos ver cómo está funcionando este proyecto en zonas rurales. Esta investigación busca analizar qué tan bien funciona el servicio VoIP en la red inalámbrica que se instaló en Huacaspata. Es importante entender cómo mejorar la calidad del servicio, viendo cómo se mueve la gente y cómo usa el Internet, para que las telecomunicaciones funcionen mejor en zonas parecidas.

1.2. Formulación del problema de investigación

¿Cómo los diferentes códecs en los modelos de tráfico VoIP influyen en la asignación de recursos de la red inalámbrica de banda ancha necesarios para ofrecer servicios de telefonía móvil y fija en las localidades rurales beneficiadas del distrito de Huacaspata, Región La Libertad?

1.3. Hipótesis

Si usamos diferentes códecs en los modelos de tráfico VoIP, entonces existe una diferencia significativa en la asignación de recursos de la red inalámbrica de banda ancha necesarios para ofrecer servicios de telefonía móvil y fija en las localidades rurales beneficiadas del distrito de Huacaspata, Región La Libertad.

1.4. Objetivos de la investigación

1.4.1. Objetivo General

Evaluar los diferentes códecs en los modelos de tráfico VoIP en la asignación de recursos de la red inalámbrica de banda ancha necesarios para ofrecer servicios de telefonía móvil y fija en las localidades rurales beneficiadas del distrito de Huacaspata, Región La Libertad.

1.4.2. Objetivos Específicos

1. Realizar un diagnóstico y estimación de la demanda de servicios de telefonía móvil y fija en las localidades rurales del distrito de Huacaspata, identificando las necesidades de conectividad y los patrones de uso actuales.
2. Evaluar el impacto de diferentes códecs en los modelos de tráfico VoIP sobre el ancho de banda requerido en la red inalámbrica de banda ancha del distrito de Huacaspata.
3. Analizar los efectos de los diferentes códecs en los modelos de tráfico VoIP en métricas clave como latencia, jitter y pérdida de paquetes en la red inalámbrica de banda ancha del distrito de Huacaspata.

Capítulo II: Marco Teórico

2.1. Antecedentes

Los antecedentes de la investigación sobre VoIP en redes de banda ancha evidencian la evolución y los desafíos actuales en este campo. Nedyalkov(2023) destaca la importancia de las plataformas de modelado de redes IP, demostrando que herramientas como GNS3 son alternativas efectivas y económicas para el estudio de estas redes.

En el contexto de la pandemia, Adhilaksono y Setiawan(2022) identificó las métricas más relevantes para evaluar la calidad de las aplicaciones VoIP, siendo las más preferidas en términos de QoE la puntuación media de opinión (MOS), la evaluación perceptual de la calidad del habla (PESQ) y la relación señal-ruido máxima (PSNR). Para QoS, destacan la fluctuación, el ancho de banda, la pérdida de paquetes y el rendimiento.

M. Mauro et al.(2023) profundizó en la predicción del comportamiento del tráfico VoIP en entornos móviles, mientras que M. D. Mauro y Liotta(2020) realizó una evaluación detallada del rendimiento del tráfico VoIP en entornos LTE-A reales, identificando el Valor Extremo Generalizado (GEV) como la distribución óptima para caracterizar parámetros críticos como jitter y RTT.

Burbur et al.(2022) abordó las limitaciones en la calidad de servicio de las redes UMTS, mientras que Olorunnisola et al.(2023) propuso una técnica innovadora que combina compresión y agregación de paquetes para mejorar el rendimiento de VoIP sobre redes de malla inalámbricas, logrando una mejora del 56.25% en el rendimiento de la red.

Los estudios de Strzeciwilk(2021) demostraron la importancia de la selección de códecs en la calidad de transmisión de voz. En el contexto de infraestructura móvil, Manap et al.(2020) analizó el desempeño de servicios celulares 4G, identificando factores críticos como la pérdida de propagación y la degradación de la señal.

Maslen y Nunez(2021) abordó la problemática de la conectividad en zonas rurales de Estados Unidos, proporcionando una base para la investigación y adaptación de redes 4G

LTE, lo cual es particularmente relevante para estudios en entornos rurales similares al distrito de Huacaspata.

Además Estela Perez y Jimenez Requejo(2024) se desarrollo una investigación de una red de banda ancha rural diseñada a medida de las necesidades de cada centro poblado en el distrito de Shamboyacu, con tecnologías inalámbricas para conectividad amplia y estable; usaron simulador NS3 para validar el rendimiento y NetAnim para visualizar el tráfico, incluyó una recopilación de datos sobre la población, viviendas e instituciones públicas, junto con una modelación detallada del tráfico según los tipos de usuarios, resultando en un modelo replicable que superó las expectativas en términos de rendimiento y adaptabilidad.

2.2. Bases Teóricas

2.2.1. Análisis de Población del distrito de Huacaspata

Usando la información de los censos nacionales del 2017(INEI, 2018) hemos obtenido la información de población, vivienda y tiempo de viaje de cada localidad a la capital de distrito Huacaspata. Es importante analizar esta información disponible para entender como se distribuye la población, vivienda en este distrito, además de conocer que tan alejados están las localidades de la capital. De un total de 66 localidades registrados en el conjunto de datos, encontramos que la población tiene mucha variabilidad, con localidades deshabitadas de 0 personas y otras muy pobladas de casi 1200. La media es de 95 personas pero la mediana de 41 muestra que hay más localidades pequeñas. Las viviendas siguen una distribución similar a la población. El tiempo de viaje también es variable, con lugares adyacentes de 0 mins y otros lejanos de más de 5 horas. La media es de 80 mins pero lo más frecuente es tiempos alrededor de 1 hora.

El análisis más detallado de los resultados mostrados en la Tabla 2.1 nos indica que la media de la población es de 93.29 sugiere que, en promedio, las localidades tienen alrededor de 93 habitantes. La desviación estándar es muy alta (165.89), lo que indica una gran variabilidad en la cantidad de población entre las localidades. El rango va desde una localidad con solo 2 habitantes hasta una con 1197, lo que muestra una disparidad

significativa en la distribución de la población. Los percentiles indican que el 25 % de las localidades tienen 17 habitantes o menos, la mediana está en 41 habitantes, y el 75 % de las localidades tienen 88 habitantes o menos, confirmando la concentración de localidades con poblaciones pequeñas. El caso de la vivienda es bastante similar a la población en general con una media es de 30.14 viviendas por localidad. Una desviación estándar de 54.13 también sugiere una gran variabilidad en el número de viviendas. El mínimo es de una sola vivienda, mientras que el máximo es de 389 viviendas. El 25 % de las localidades tienen 6 o menos viviendas, y el 75 % tienen 31 o menos, indicando que la mayoría de las localidades tienen un número relativamente bajo de viviendas. El caso del tiempo de viaje tiene una media de 96.24 minutos. La desviación estándar de 72.23 muestra una amplia gama de tiempos de viaje entre las localidades. Un mínimo de 0 indica el punto de referencia que es la capital de distrito. El 25 % de las localidades tienen tiempos de viaje de 41 unidades o menos, y el 75 % tienen tiempos de viaje de 122.5 unidades o menos, lo que sugiere que mientras algunas localidades son accesibles relativamente rápido, otras están bastante alejadas.

Tabla 2.1

Análisis descriptivo: Localidades de Huacaspata

	Población	Vivienda	Tiempo de viaje
Cantidad	63	63	63
Promedio	93.29	30.14	96.24
Desviación estándar	165.89	54.13	72.23
mínimo	2	1	0
25 %	17	6	41
50 %	41	14	82
75 %	88	31	122.5
máximo	1197	389	317

En general, estos resultados reflejan una población y un número de viviendas con una distribución muy dispersa y sesgada hacia localidades más pequeñas y menos desarrolladas en cuanto al número de viviendas. El tiempo de viaje también varía considerablemente, lo que podría afectar el acceso a servicios esenciales como la telecomunicación, particu-

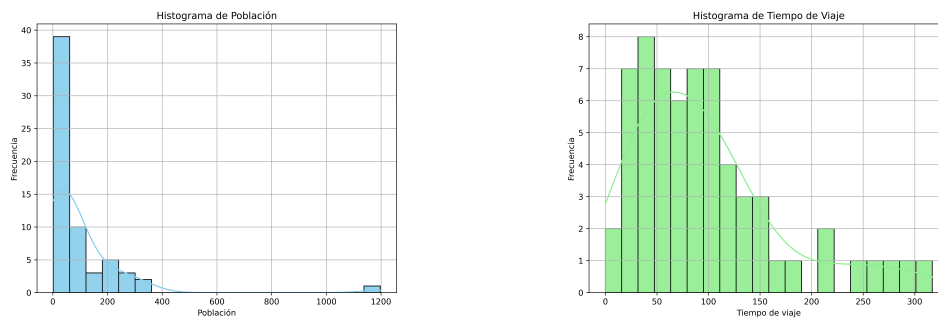
larmente en localidades con tiempos de viaje más largos.

Figura 2.1

Análisis de Población y Tiempo de viaje del distrito de Huacaspata

(a) *Histograma de Población con densidad.*

(b) *Histograma de Tiempo de viaje con densidad*



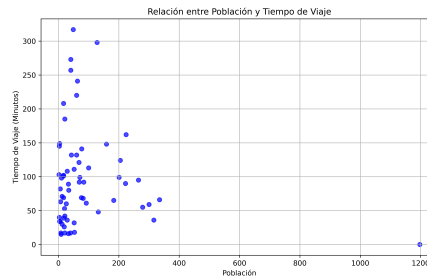
La distribución de la población muestra una concentración significativa de localidades con poblaciones pequeñas y una larga cola hacia valores más altos, indicando algunas localidades con poblaciones mucho mayores. La curva de densidad resalta esta asimetría (Ver Fig. 2.1a). La distribución de vivienda presenta un patrón similar, con la mayoría de las localidades teniendo un número pequeño de viviendas, pero algunas localidades destacan por tener muchas más viviendas.

La distribución del tiempo de viaje también muestra una dispersión considerable, con la mayoría de las localidades agrupadas en tiempos de viaje más cortos y algunas extendiéndose hacia tiempos más largos (Ver Fig. 2.1).

El gráfico de dispersión entre la población y el tiempo de viaje, en la Fig. 2.2 enfatiza la gran dispersión entre estos datos, sin una correlación clara y directa. Esto sugiere que la accesibilidad a servicios como las de telecomunicaciones puede no estar directamente relacionada con el tamaño de la población de una localidad, destacando la importancia de considerar otros factores, como la geografía o la infraestructura existente, al planificar la expansión de servicios en el distrito de Huacaspata.

Figura 2.2

Gráfico de Dispersión de Población y Tiempo de viaje



2.2.1.1. Localidades beneficiadas

En el contexto de este estudio, se han seleccionado trece localidades del distrito de Huacaspata, en la provincia de Pataz, aplicando los criterios de ruralidad establecidos por Pronatel. Estos criterios incluyen: poblaciones con más de 100 habitantes, la presencia de instituciones públicas tales como escuelas de nivel primario o secundario, centros de salud o comisarías, y que, en esta primera fase, exista línea de vista directa desde la capital del distrito de Huacaspata. La selección de estas localidades se detalla en la Tabla [2.2](#).

Tabla 2.2

Localidades Beneficiadas usando los criterios de ruralidad de Pronatel

NRO	Localidad	Población	Vivienda	Colegios	Centro de Salud	Otras
1	HUANCASPATA	1197	389	1	0	1
2	CHOCOBAMBA	300	84	1	1	0
3	PATRAMARCA	316	98	1	0	1
4	FORTALEZA ANDINA	132	34	1	0	0
5	COYARTUNA	183	54	1	0	0
6	BUENOS AIRES	265	86	1	1	0
7	PUEBLO LIBRE	222	79	1	1	0
8	SOL DE VILLA	159	36	1	1	0
9	COCHACARA	224	71	1	1	0
10	WILCAYACO	205	66	1	1	0
11	MOLLEPAMPA	335	137	1	1	0
12	SHIHUANTO	128	32	1	1	0
13	PACOBAMBA	201	56	1	1	0

Esta información es fundamental para estimar la demanda de servicios de telecomunicaciones en la zona, no solo en lo referente al tráfico de voz, utilizando distintos códecs

de VoIP, sino también en cuanto al tráfico de internet, tanto móvil como fijo. El análisis se centrará en el desempeño de la red, con especial atención en el tráfico de VoIP sobre una infraestructura inalámbrica que se extiende desde la capital del distrito.

Para la estimación de la demanda se considerarán dos tipos principales de servicios:

1. Servicio de Internet: Este comprende una conexión de alta velocidad, conforme a lo establecido en la Resolución Ministerial 1197-2022-MTC y adaptado a las tasas de penetración regional publicadas por Osiptel. Se diferenciará entre:
 - a) Móvil: Basado en la penetración de usuarios móviles y la población de cada localidad.
 - b) Fijo: Calculado según el número de viviendas en la localidad.
2. Servicios de Telefonía (VoIp): Se analizará el tráfico de voz sobre Ip, diferenciando entre:
 - a) Móvil: Estimado a partir de la población de cada localidad.
 - b) Fijo: Basado en el número de viviendas.

Adicionalmente, se incorporará la demanda institucional siguiendo las recomendaciones del MTC. Para el tráfico telefónico, ya sea móvil o fijo, se calculará primero en Erlangs para luego, con un grado de servicio típico de 0.01, estimar el número de troncales necesarias, según se detalla en la Tabla [2.3](#), tenemos la demanda tanto de Internet como de telefonía VoIP de cada localidad.

2.2.2. Modelos de tráfico VoIP fijo y móvil

2.2.2.1. Codecs de Voip

Los códecs de VoIP representan una evolución en la manera en que transmitimos la voz a través de las redes digitales. En lo más alto encontramos al G.711, que con sus robustos 64 Kbps y una carga útil de 160 bytes, tiene la mejor calidad con una puntuación MOS

Tabla 2.3*Demanda Estimada Total: Considerando codec G711*

Nro	Localidad	Circuitos Necesarios para Telefonía				Vtx(Mbps)		Capacidad Estimada (Mbps)		
		Móvil		Fija		Descarga	Subida	Descarga	Subida	Agregado
		Entrantes	Salientes	Entrantes	Salientes					
1	HUANCASPATA	20	24	4	4	2.1	2.5	373.1	162.5	535.6
2	CHOCOBAMBA	8	9	2	2	0.9	1.0	118.9	64.0	182.8
3	PATRAMARCA	8	10	2	2	0.9	1.1	116.9	62.1	178.9
4	FORTALEZA ANDINA	5	6	2	2	0.6	0.7	66.6	35.7	102.3
5	COYARTUNA	6	7	2	2	0.7	0.8	81.7	41.8	123.5
6	BUENOS AIRES	7	9	2	2	0.8	1.0	108.8	60.0	168.8
7	PUEBLO LIBRE	6	8	2	2	0.7	0.9	98.7	55.9	154.6
8	SOL DE VILLA	5	6	2	2	0.6	0.7	88.6	51.7	140.3
9	COCHACARA	6	8	2	2	0.7	0.9	98.7	55.9	154.6
10	WILCAYACO	6	7	2	2	0.7	0.8	98.7	55.8	154.5
11	MOLLEPAMPA	8	10	2	3	0.9	1.1	143.9	73.1	217.0
12	SHIHUANTO	5	6	1	2	0.5	0.7	78.5	47.7	126.2
13	PACOBAMBA	6	7	2	2	0.7	0.8	93.7	53.8	147.5

de 4.1, la más elevada entre sus pares, pero viene acompañada de una mayor demanda de recursos de red.

En el otro extremo del espectro, el G.729 con una alta eficiencia. Con apenas 8 Kbps y una modesta carga útil de 20 bytes, logra mantener una calidad sorprendentemente alta con un MOS de 3.92.

Entre estos extremos, encontramos la familia G.726, que ofrece dos variantes interesantes. La versión de 32 Kbps se presenta como un equilibrado mediador, transportando 80 bytes de carga útil y manteniendo un respetable MOS de 3.85. Su hermano menor, la variante de 24 Kbps, reduce la carga a 60 bytes sin sacrificar demasiado en calidad, sosteniendo un MOS de 3.8.

Finalmente, el G.728 adopta una estrategia única. Con sus 16 Kbps y 60 bytes de carga útil, se toma un poco más de tiempo para procesar cada paquete, extendiendo el intervalo a 30 ms, a diferencia de los 20 ms estándar de sus compañeros. Aunque su MOS de 3.61 es el más modesto del grupo, sigue ofreciendo una calidad más que aceptable para las comunicaciones de voz. Todos estos códecs comparten un rasgo común: el compromiso entre la calidad de la voz y el consumo de recursos de red. Es como una orquesta donde cada instrumento tiene su papel específico, y la elección del códec adecuado dependerá de la sinfonía particular que necesitemos interpretar en nuestra red.

Tabla 2.4*Codecs VoIP usados en Simulación*

Codec	Vtx(Kbps)	Carga Util(Bytes)	Intervalo(ms)	MSO
G.711	64	160	20	4.1
G.729	8	20	20	3.92
G.726	32	80	20	3.85
G.726	24	60	20	3.8
G.728	16	60	30	3.61

2.2.2.2. Modelos de Poisson y Markov

Los modelos de Poisson y Markov fueron utilizados inicialmente para modelar el tráfico en redes telefónicas tradicionales. En estos modelos, las llegadas de llamadas siguen una distribución de Poisson, mientras que los tiempos de servicio (duración de las llamadas) se asumen con distribución exponencial. Esto lleva a un proceso de nacimiento y muerte que puede ser representado por una cadena de Markov de estados finitos.

Si bien estos modelos fueron adecuados para las redes telefónicas, presentan limitaciones para representar fielmente el tráfico en redes modernas de voz sobre IP (VoIP). Esto se debe a que el tráfico VoIP exhibe características más complejas, como la variabilidad en los tiempos entre llegadas de paquetes y la correlación entre ellos, lo que se conoce como autosimilitud.

2.2.2.3. Modelo de Fuente On/Off

El modelo de fuente On/Off es ampliamente utilizado para representar el comportamiento de una llamada VoIP individual. En este modelo, la fuente alterna entre dos estados:

- Estado On: representa los períodos de actividad de la voz, donde se genera tráfico a una tasa constante.
- Estado Off: representa los períodos de silencio, donde no se genera tráfico.

Los tiempos de permanencia en cada estado se asumen independientes e idénticamente distribuidos (IID). Comúnmente se utiliza una distribución exponencial para modelar estos

tiempos, lo que lleva a un proceso de Markov de dos estados. Sin embargo, también es posible utilizar distribuciones de cola pesada, como la Pareto, para capturar la variabilidad observada en el tráfico real.

La fuente On/Off básica se caracteriza por la tasa de generación de paquetes en el estado activo, y por las tasas de transición entre los estados. Con estos parámetros se pueden obtener métricas como la probabilidad de estado estacionario, el número promedio de paquetes generados por unidad de tiempo, entre otras.

2.2.2.4. Multiplexación de Fuentes On/Off

En una red VoIP, múltiples llamadas comparten los recursos de transmisión. Por lo tanto, es necesario considerar la multiplexación de varias fuentes On/Off para obtener el tráfico agregado.

Al combinar un número suficiente de fuentes On/Off independientes, se obtiene un proceso con características distintivas. El tráfico agregado exhibe autosimilitud, es decir, presenta patrones estadísticos similares en diferentes escalas de tiempo. Esto se traduce en una alta variabilidad y en la presencia de ráfagas en el tráfico.

Desde un punto de vista matemático, el tráfico agregado puede ser modelado mediante procesos estocásticos fractales, como el movimiento Browniano fractal (fBm). Estos procesos permiten capturar la dependencia de rango largo y la autosimilitud observadas.

Alternativamente, se pueden utilizar modelos de Markov más complejos, como las cadenas de Markov moduladas por procesos de Poisson (MMPP) o las cadenas de Markov moduladas por procesos de renovación (MMRP), para representar la multiplexación de fuentes On/Off. Estos modelos permiten una caracterización más detallada del tráfico, a costa de una mayor complejidad computacional.

2.2.2.5. Modelos de Pareto

Los modelos basados en la distribución de Pareto han ganado popularidad para representar el tráfico autosimilar en redes modernas, incluyendo el tráfico VoIP. La distribución de Pareto es una distribución de cola pesada, lo que significa que tiene una alta probabilidad

de generar valores extremos.

En el contexto de VoIP, los modelos de Pareto se utilizan para describir la distribución de los tiempos entre llegadas de paquetes y las duraciones de las ráfagas de tráfico. La función de densidad de probabilidad de la distribución de Pareto viene dada por:

$$f(x) = (a * b^a) / x^{a+1}, \text{ para } x \geq b$$

donde “a” es el parámetro de forma y “b” es el parámetro de escala.

Al ajustar los parámetros de la distribución, se pueden obtener diferentes grados de variabilidad y autosimilitud en el tráfico generado. Un valor de “a” cercano a 1 indica un alto grado de autosimilitud, mientras que valores mayores de “a” corresponden a un tráfico más suave.

Los modelos de Pareto son útiles para generar tráfico sintético con características similares al tráfico VoIP real, lo que permite evaluar el desempeño de la red y dimensionar adecuadamente los recursos necesarios.

Además, los parámetros de la distribución de Pareto pueden ser estimados a partir de trazas de tráfico real, utilizando técnicas estadísticas como el método de máxima verosimilitud o el análisis de la varianza agregada.

En resumen, los modelos de Pareto proporcionan una herramienta flexible y eficiente para modelar el tráfico VoIP autosimilar, capturando su variabilidad y su naturaleza en ráfagas.

2.2.3. Software de Planificación y Simuladores

Para el desarrollo de las comunicaciones inalámbricas, el proceso de diseño y optimización de redes requiere un conjunto sofisticado de herramientas que permitan predecir, analizar y validar el comportamiento de los sistemas antes de su implementación física. En este contexto, dos categorías de herramientas emergen como pilares fundamentales: los planificadores de RF y los simuladores de red.

Los planificadores de RF han transformado radicalmente la manera en que conceptualizamos y optimizamos las redes inalámbricas. Estas herramientas de software, ejemplificadas por sistemas como Radio Mobile y RadioPlanner y LinkPlanner, proporcionan

un conjunto integral de funcionalidades que automatizan y mejoran significativamente el proceso de planificación. Su verdadero poder radica en la capacidad para modelar la propagación de ondas de radio, generar mapas de cobertura detallados y realizar análisis exhaustivos del presupuesto de enlace, convirtiéndolos en recursos indispensables para los ingenieros de telecomunicaciones.

La característica más destacada de estos planificadores es su capacidad para simular y predecir el comportamiento de las señales de radio en diversos escenarios. Mediante la implementación de sofisticados modelos de propagación, estas herramientas pueden anticipar la intensidad de la señal en cualquier punto de la zona de cobertura, permitiendo identificar potenciales áreas problemáticas antes de la implementación física de la red.

Mientras los planificadores de RF se centran en los aspectos físicos de la propagación de señales, los simuladores de red emergen como herramientas complementarias fundamentales para evaluar el comportamiento del tráfico y la operación de los protocolos de red. Estos simuladores permiten a los ingenieros experimentar con diferentes configuraciones y escenarios sin la necesidad de implementar físicamente la infraestructura, pero usando los resultados de comportamiento de las señales de RF que establece límites razonables y justificados que permite la simulación más realista lo que resulta en significativos ahorros de tiempo y recursos.

En el ecosistema de simuladores, destacan herramientas como NS-3, que ha ganado considerable popularidad en entornos académicos y de investigación. Su arquitectura basada en C++ y su diseño flexible permiten a los usuarios desarrollar modelos personalizados, facilitando tanto la investigación avanzada como el aprendizaje de conceptos fundamentales de redes. Aunque ya discontinuado, OPNET también dejó un importante legado con su extensa biblioteca de modelos predefinidos y su capacidad para realizar pruebas detalladas en diversos tipos de redes.

Una mención especial merece Emulab, que representa un enfoque híbrido entre la simulación pura y las implementaciones físicas. Esta plataforma permite realizar pruebas en condiciones que se aproximan significativamente a los escenarios reales, proporcionando

un puente valioso entre el mundo de la simulación y las implementaciones prácticas.

La combinación de planificadores de RF y simuladores de red proporciona a los ingenieros un conjunto completo de herramientas para el diseño, análisis y optimización de redes modernas. Mientras los planificadores de RF garantizan una cobertura y propagación de señal óptimas, los simuladores permiten validar el comportamiento de la red bajo diferentes condiciones de tráfico y configuraciones, resultando en diseños más robustos y eficientes.

2.2.3.1. El Presupuesto de Enlace

El presupuesto de enlace constituye uno de los cálculos más críticos en la planificación de redes inalámbricas. Este análisis determina la viabilidad de un enlace de radio al considerar todos los factores que afectan a la señal en su trayecto desde el transmisor hasta el receptor. Para realizar estos cálculos, los planificadores de RF implementan diferentes modelos de propagación, cada uno adaptado a escenarios específicos:

- **Modelo de Pérdidas en Espacio Libre** Este modelo fundamental proporciona una base teórica para entender la propagación de señales en condiciones ideales. La fórmula $FSL(dB) = 32,4 + 20\log_{10}(D) + 20\log_{10}(F)$, donde D es la distancia en kilómetros y F la frecuencia en MHz, permite calcular las pérdidas básicas que experimenta una señal al atravesar el espacio libre.
- **Modelo de Dos Rayos** Construyendo sobre el modelo básico, el modelo de dos rayos incorpora el efecto de la reflexión terrestre, proporcionando predicciones más precisas para enlaces con línea de vista directa sobre superficies reflectantes. Este modelo resulta particularmente útil en entornos donde la reflexión juega un papel significativo en la propagación de la señal.
- **Modelo Longley-Rice** Para escenarios más complejos, el modelo Longley-Rice emerge como la opción más sofisticada. Este modelo considera múltiples factores ambientales, incluyendo la difracción causada por obstáculos naturales y artificiales, proporcionando predicciones más precisas en entornos reales.

2.2.3.2. Factores Adicionales en la Planificación

Los planificadores modernos de RF no se limitan al análisis de la propagación básica, sino que incorporan consideraciones adicionales cruciales para un diseño de red exitoso:

- Las pérdidas en la línea de alimentación y conectores, que pueden afectar significativamente el rendimiento del sistema.
- La absorción atmosférica, especialmente relevante en frecuencias más altas.
- Las pérdidas en filtros y duplexores, componentes esenciales en sistemas de comunicación complejos.

2.2.3.3. Proceso Integral de Planificación

El proceso de planificación de RF se desarrolla en fases bien definidas:

- Dimensionamiento inicial, donde se establecen los requisitos básicos de cobertura y capacidad.
- Preplanificación, que incluye la selección preliminar de sitios y la definición de objetivos específicos.
- Planificación detallada, donde se realizan los cálculos precisos del presupuesto de enlace y se optimizan los parámetros de red.

2.2.3.4. Radio Mobile: La Alternativa de Código Abierto

Radio Mobile, desarrollado por Roger Coudé (VE2DBE), representa una solución gratuita y accesible para la planificación de redes de radio. A pesar de su origen en la comunidad de radioaficionados, ha ganado reconocimiento en aplicaciones profesionales gracias a sus capacidades fundamentales:

- Implementación del modelo ITM (Irregular Terrain Model) Longley-Rice para enlaces obstruidos
- Análisis de presupuesto de enlace con consideración de múltiples variables

- Generación de mapas de cobertura en formatos polares y cartesianos
- Capacidades de análisis de rutas para sistemas móviles Interfaz gráfica intuitiva con soporte para mapas de alta resolución

2.2.3.5. RadioPlanner 3.0

RadioPlanner 3.0 emerge como una solución comercial robusta, desarrollada por expertos con décadas de experiencia en el sector. Sus características destacadas incluyen:

- Versatilidad en Tipos de Red
 - Redes móviles (UMTS, GSM, WCDMA, P25, TETRA)
 - Redes IoT LPWAN
 - Sistemas de radiodifusión terrestre
 - Comunicaciones aire-tierra
- Capacidades Analíticas Avanzadas
 - Estudios completos de área para múltiples parámetros
 - Análisis de potencia recibida bidireccional
 - Evaluación de rendimiento y cobertura
 - Herramientas específicas para radiodifusión
- Modelos de Propagación Diversos
 - ITU-R P.1812-6 para servicios terrestres
 - Longley-Rice para América del Norte
 - Okumura-Hata para entornos urbanos
 - Modelos específicos para LTE y 5G

2.2.3.6. LinkPlanner

LinkPlanner, desarrollado por Cambium Networks, se destaca como la herramienta más adecuada para investigaciones especializadas en enlaces punto a punto y redes PMP. Sus características específicas lo convierten en una opción superior para análisis detallados:

- Modelado de Propagación Avanzado LinkPlanner implementa modelos de propagación sofisticados que incluyen:
 - ITU-R P.530 para frecuencias superiores a 10 GHz
 - ITU-R P.526 para cálculos de disponibilidad
 - Modelo Vigants-Barnett para análisis de multitrayecto
- Gestión de Clutter El software incorpora un sistema avanzado de gestión de clutter que permite:
 - Representación visual mediante bloques de colores
 - Configuración personalizada de alturas de clutter
 - Integración en cálculos de pérdida de señal
- Capacidades de Planificación Especializadas
 - Diseño detallado de enlaces punto a punto
 - Análisis comprehensivo de redes PMP
 - Herramientas de sincronización TDD
 - Soporte para diversidad espacial y de frecuencia
- Funcionalidades para Gestión de Proyectos
 - Generación automática de presupuestos de enlace
 - Creación de listas de materiales (BOM)
 - Informes detallados de propuestas e instalación
 - Análisis de costes y recursos

2.2.3.7. Fundamentos del Simulador NS3

El simulador NS3 representa una evolución significativa en el campo de la simulación de redes. Este software de código abierto, desarrollado específicamente para la investigación y educación en redes de comunicación, se distingue por su arquitectura modular y su capacidad para modelar con precisión diversos escenarios de red. A diferencia de su predecesor NS2, NS3 fue completamente rediseñado utilizando C++ moderno, lo que permite una mayor flexibilidad y rendimiento en las simulaciones.

La arquitectura de NS3 se fundamenta en un sistema modular que, facilita la extensión y personalización del simulador. El núcleo del sistema se compone de varios elementos fundamentales:

- El motor de simulación de eventos discretos
- El sistema de gestión de objetos y atributos
- Las bibliotecas de helpers para configuración de red
- Los módulos de red que implementan diferentes protocolos y tecnologías

Esta organización modular permite a los investigadores construir escenarios complejos de manera incremental, agregando componentes según las necesidades específicas de cada simulación.

2.2.3.8. Simulación de VoIP en NS3

La simulación de tráfico VoIP en NS3 requiere una comprensión profunda de los modelos de tráfico y las características específicas de las comunicaciones de voz. El tráfico VoIP se puede modelar eficazmente utilizando procesos On/Off, donde los períodos de actividad e inactividad siguen distribuciones probabilísticas específicas que reflejan los patrones naturales del habla.

2.2.3.9. Generación de Tráfico y Análisis

El proceso de generación y análisis de tráfico en NS3 involucra múltiples componentes: La generación de tráfico utiliza generadores aleatorios basados en diferentes distribuciones

estadísticas. Estos generadores son fundamentales para simular:

- Patrones de llegada de paquetes
- Variaciones en el tamaño de los paquetes
- Comportamiento de las ráfagas de tráfico
- Errores y pérdidas en la transmisión

2.2.3.10. Métricas y Evaluación de Rendimiento

NS3 proporciona herramientas robustas para la recolección y análisis de métricas de rendimiento cruciales en las comunicaciones VoIP:

- Latencia end-to-end
- Jitter y variación del retardo
- Pérdida de paquetes
- Throughput efectivo
- Calidad de servicio (QoS)

Capítulo III: Marco Metodológico

3.1. Introducción

El contraste de hipótesis es una etapa fundamental en el proceso de investigación, ya que permite evaluar la validez de las afirmaciones propuestas en relación con el problema de estudio. En esta investigación, el enfoque se centra en determinar cómo los diferentes códecs empleados en los modelos de tráfico VoIP influyen en la asignación de recursos de las redes inalámbricas de banda ancha, con el propósito de garantizar servicios de calidad para telefonía fija y móvil en las localidades rurales del distrito de Huacaspata, región La Libertad.

3.1.1. Tipo de Investigación

Este estudio es de tipo explicativo. Busca no solo describir las características del fenómeno, sino también analizar las relaciones causales entre las variables principales: los códecs de VoIP (variable independiente) y la asignación de recursos en la red inalámbrica, evaluada a través de métricas clave como ancho de banda, latencia, jitter y pérdida de paquetes (variables dependientes). Este tipo de investigación es adecuado para entender cómo el uso de diferentes códecs impacta en el desempeño de la red, lo que tiene implicaciones directas para la planificación y optimización de infraestructura en áreas rurales.

3.1.2. Diseño de Investigación

El diseño adoptado es no experimental, transversal y comparativo:

3.1.2.1. No experimental

No se manipularán deliberadamente las variables independientes (los códecs), sino que se observarán y analizarán sus efectos mediante simulaciones controladas. Esto es consistente con la naturaleza del problema, dado que se busca replicar condiciones reales de uso en redes de banda ancha.

3.1.2.2. Transversal

Los datos se recopilarán en un único momento temporal, analizando las métricas obtenidas para cada códec bajo las mismas condiciones simuladas. Este enfoque permite realizar una comparación directa entre los resultados generados por los diferentes escenarios.

3.1.2.3. Comparativo

El diseño se centra en la comparación de métricas de QoS entre diferentes códecs, como G711, G729, G726 y G728. Esto incluye identificar cuál ofrece un mejor balance entre calidad de servicio y consumo de recursos en un contexto rural.

3.1.2.4. Justificación del Diseño

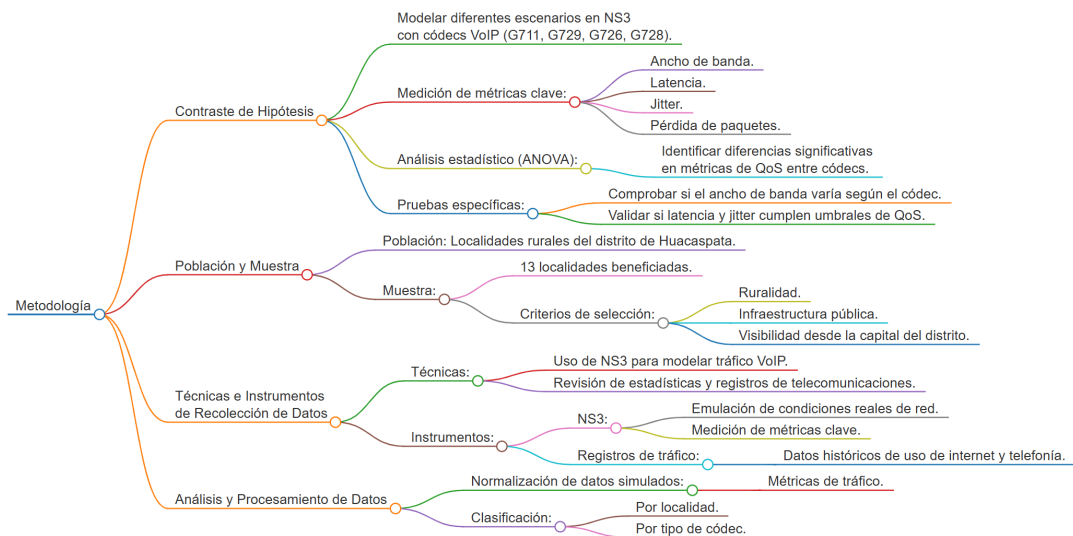
El diseño no experimental y transversal es el más apropiado para este estudio porque:

- Permite trabajar con datos simulados en un entorno controlado, lo que asegura que las diferencias observadas sean atribuibles únicamente a los códecs utilizados.
- Facilita la comparación de múltiples métricas de QoS de forma simultánea y bajo un mismo marco temporal, asegurando la consistencia en las condiciones de evaluación.
- Proporciona resultados inmediatos que pueden ser utilizados para proponer estrategias de optimización en la asignación de recursos de red en áreas rurales.

3.2. Contraste de Hipótesis

La hipótesis planteada sostiene que los diferentes códecs en los modelos de tráfico VoIP impactan significativamente en la asignación de recursos de la red inalámbrica de banda ancha en el distrito de Huacaspata. Para contrastar esta hipótesis, seguiremos los siguientes pasos:

1. Se realizarán simulaciones utilizando NS3, modelando diferentes escenarios con los códecs VoIP más representativos (G711, G729, G726 y G728).
2. Cada simulación incluirá la medición de métricas clave como ancho de banda, latencia, jitter y pérdida de paquetes.

Figura 3.1*Proceso Metodológico de Investigación*

3. Se aplicará análisis estadístico (ANOVA) para identificar diferencias significativas en las métricas de QoS entre los diferentes códecs.

Pruebas específicas:

1. Comprobar si el ancho de banda requerido varía significativamente según el códec.
2. Validar si métricas como latencia y jitter cumplen los umbrales de QoS con cada códec.

3.3. Población y Muestra

La población del estudio incluye las localidades rurales del distrito de Huacaspata

Muestra: 13 localidades beneficiadas. Criterios de selección: Se basan en la ruralidad, la presencia de infraestructura pública y la visibilidad desde la capital del distrito.

3.4. Técnicas e Instrumentos de Recolección de Datos

Para evaluar los códecs y sus efectos:

3.4.1. Técnicas

- Simulación computacional: Uso del simulador NS3 para modelar tráfico VoIP.
- Estudio documental: Revisión de estadísticas y registros locales de demanda de servicios de telecomunicaciones.

3.4.2. Instrumentos

- NS3: Para emular condiciones reales de red y medir métricas clave.
- Registros de tráfico: Datos históricos de uso de internet y telefonía, si están disponibles.

3.5. Análisis y Procesamiento de Datos

El análisis será descriptivo y analítico.

1. Procesamiento inicial:

- Normalización de datos simulados (por ejemplo, métricas de tráfico).
- Clasificación por localidad y tipo de códec.

2. Análisis descriptivo:

- Tablas y gráficos para visualizar diferencias en consumo de recursos según el códec.
- Estadísticas clave como media, desviación estándar y percentiles para métricas de QoS.

3. Análisis inferencial:

- Uso de ANOVA para comparar métricas entre códecs.
- Pruebas post hoc (Tukey) para identificar diferencias específicas.

4. Interpretación final

- Relación entre los códecs y la experiencia del usuario (QoS).
- Recomendaciones para optimizar el uso de recursos según las condiciones de Huacaspata.

Capítulo IV: Resultados

4.1. Estimación de Demanda

En esta investigación estamos interesados en analizar el tráfico VoIP con diferentes codecs por lo que en la Tabla 4.1 tenemos solo este tráfico con diferentes codecs que han sido seleccionados con MSO mayores a 3.8

Tabla 4.1

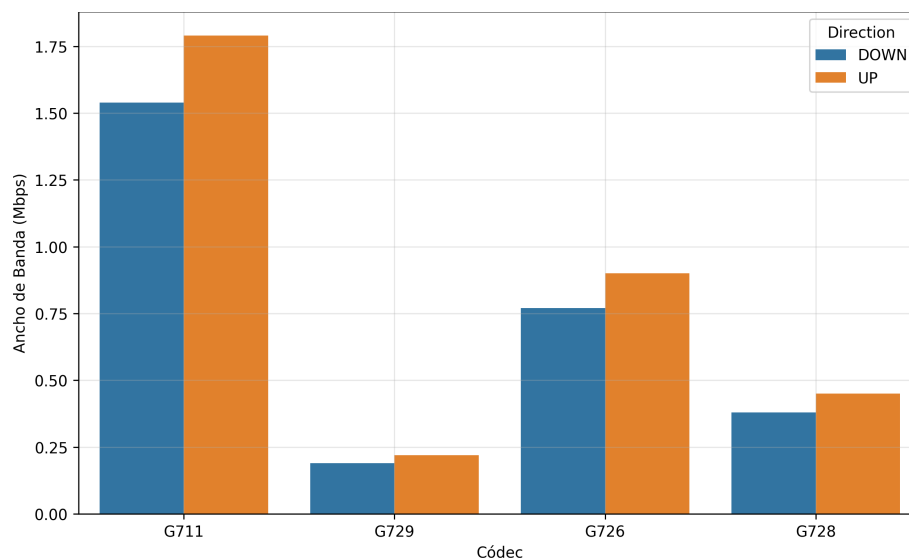
Capacidad Estimada de Tráfico VoIP con Diferentes Codecs

NRO	LOCALIDAD	Ctd Usuarios		G711 (Mbps)		G729(Mbps)		G726(Mbps)		G728(Mbps)	
		IN	OUT	DOWN	UP	DOWN	UP	DOWN	UP	DOWN	UP
1	HUANCASPATA	24	28	1.54	1.79	0.19	0.22	0.77	0.90	0.38	0.45
2	CHOCOBAMBA	10	11	0.64	0.70	0.08	0.09	0.32	0.35	0.16	0.18
3	PATRAMARCA	10	12	0.64	0.77	0.08	0.10	0.32	0.38	0.16	0.19
4	FORTALEZA ANDINA	7	8	0.45	0.51	0.06	0.06	0.22	0.26	0.11	0.13
5	COYARTUNA	8	9	0.51	0.58	0.06	0.07	0.26	0.29	0.13	0.14
6	BUENOS AIRES	9	11	0.58	0.70	0.07	0.09	0.29	0.35	0.14	0.18
7	PUEBLO LIBRE	8	10	0.51	0.64	0.06	0.08	0.26	0.32	0.13	0.16
8	SOL DE VILLA	7	8	0.45	0.51	0.06	0.06	0.22	0.26	0.11	0.13
9	COCHACARA	8	10	0.51	0.64	0.06	0.08	0.26	0.32	0.13	0.16
10	WILCAYACO	8	9	0.51	0.58	0.06	0.07	0.26	0.29	0.13	0.14
11	MOLLEPAMPA	10	13	0.64	0.83	0.08	0.10	0.32	0.42	0.16	0.21
12	SHIHUANTO	6	8	0.38	0.51	0.05	0.06	0.19	0.26	0.10	0.13
13	PACOBAMBA	8	9	0.51	0.58	0.06	0.07	0.26	0.29	0.13	0.14

Al analizar el tráfico en las diferentes localidades del distrito de Huacaspata, tenemos un patrón sobre el comportamiento de los codecs de voz. El codec G711, con mayor precisión en la transmisión de voz, tiene un considerable consumo de ancho de banda,

En contraste, el codec G729 se revela como la solución más eficiente en términos de consumo de ancho de banda, ofreciendo una alternativa óptima para áreas donde los recursos de red son limitados. Su capacidad para mantener una calidad de voz aceptable mientras minimiza el consumo de datos lo posiciona como una opción estratégica para implementaciones rurales.

Un aspecto particularmente interesante es la consistente relación UP/DOWN de aproximadamente 1.16 a través de todos los codecs, lo que sugiere un patrón estable en la arquitectura de comunicación bidireccional. Esta proporción constante indica una estruc-

Figura 4.1*Comparación de Ancho de Banda - Promedio Localidades*

tura de tráfico bien balanceada, facilitando la predicción y gestión de recursos de red(Ver Figura 4.1).

La localidad de Huancaspata se distingue debido a que muestra los valores más elevados para todos los códecs estudiados. Esta característica podría atribuirse a una mayor densidad de usuarios o a una infraestructura de red más robusta en esta localidad.

4.2. Red de Acceso Hucaspata

El diseño de la red para la localidad de Huacaspata se realizó utilizando la versión en línea de LinkPlanner. Considerando una demanda proyectada de 30 usuarios potenciales y un tráfico agregado superior a 500 Mbps para todos los servicios, se implementó una solución basada en dos celdas para satisfacer adecuadamente esta demanda estimada. Como se puede observar en la Figura 4.3, se establecieron líneas de vista efectivas hacia los potenciales usuarios. Tras un proceso de optimización enfocado en maximizar la capacidad, se logró que la mayoría de los usuarios operaran en el modo más alto de modulación, alcanzando los siguientes resultados(Ver Figura 4.2):

- Celda 1: 137.5 Mbps de descarga y 80.88 Mbps de subida con un tráfico agregado

de 218.38 Mbps

- Celda 2: 245.16 Mbps de descarga y 111.84 Mbps de subida con un tráfico agregado de 356.99Mbps

En conjunto, el sistema proporciona una capacidad total de 382.56 Mbps de descarga y 192.73 Mbps de subida, superando la demanda estimada en 5 Mbps para descarga y aproximadamente 25 Mbps para subida. Este margen adicional resulta particularmente ventajoso para el manejo del tráfico VoIP, cuyo ancho de banda varía entre 0.3 Mbps y 1.6 Mbps dependiendo del códec utilizado. La holgura en la capacidad garantiza una operación fluida con cualquier códec seleccionado, considerando que estos se ajustan dinámicamente según los recursos disponibles de la red y las condiciones de tráfico predominantes.

4.3. Simulador con Diferentes Codec

El simulador desarrollado busca recrear un escenario realista de comunicaciones VoIP, similar a lo que podríamos encontrar en una implementación práctica. Les explico cómo funciona todo el sistema.

Tenemos varios usuarios en sus casas u oficinas queriendo hacer llamadas VoIP. Cada usuario tiene su PC conectada a una red local (usando Ethernet), que a su vez se conecta a una torre central. Esta torre es como las que vemos en los cerros o edificios altos, que después se conecta a un servidor central donde se procesa todo el tráfico de voz.

La parte interesante viene cuando analizamos los códecs disponibles. Cada códec tiene sus propias características que afectan directamente la calidad de la llamada y el consumo de recursos de red. Para entender mejor esto, vamos a analizar cada códec considerando

Figura 4.2

Reprote de Capacidad de Celdas - Localidad de Huacaspata

PMP Network Devices (2)

<input type="checkbox"/>	Name ↑	Band	Product	Country	Bandwidth	Height (m)	SM Range	Range Units	EIRP (dBm)	Connected Subscribers	Unconnecte Subscribers	Total Throughput
<input type="checkbox"/>	Network Site 001 : 054	Upper 6 GHz	PMP 450v 4x4 (preliminary)	Other	40 MHz	10.0	1,200	kilometers	42.6	17	0	218.38 Mbps
<input type="checkbox"/>	Network Site 001 : 174	Upper 6 GHz	ePMP 4600 (preliminary)	Other	40 MHz	30.0	0,740	kilometers	43.9	16	0	356.99 Mbps

Figura 4.3

Red de Acceso Localidad de Huacaspata



su sobrecarga y eficiencia real.

Primero, es importante establecer que todos los códecs tienen una sobrecarga constante compuesta por:

- IP header: 20 bytes
- UDP header: 8 bytes
- P2P/PPP overhead: 6 bytes
- Ethernet (CSMA): 18 bytes (14 bytes header MAC + 4 bytes FCS)

Lo que nos da una sobrecarga total constante de 52 bytes para cada paquete.

El G711 es como tener una llamada en HD - usa 64 Kbps y tiene la mejor calidad (MOS de 4.1). Con un payload de 160 bytes cada 20ms, alcanza una eficiencia del 75.47% (160 bytes de payload sobre un total de 212 bytes por paquete). Su tasa efectiva, considerando la sobrecarga, llega a 84.8 Kbps. Es como elegir ver Netflix en alta definición - mejor calidad pero más datos.

El G729 es mucho más eficiente en términos de ancho de banda - solo usa 8 Kbps con una calidad todavía muy aceptable (MOS de 3.92). Sin embargo, con un payload de solo 20 bytes cada 20ms, su eficiencia cae al 27.78 % (20 bytes sobre 72 bytes totales), con una tasa efectiva de 28.8 Kbps. A pesar de su baja eficiencia, sigue siendo muy útil en redes congestionadas por su bajo consumo total.

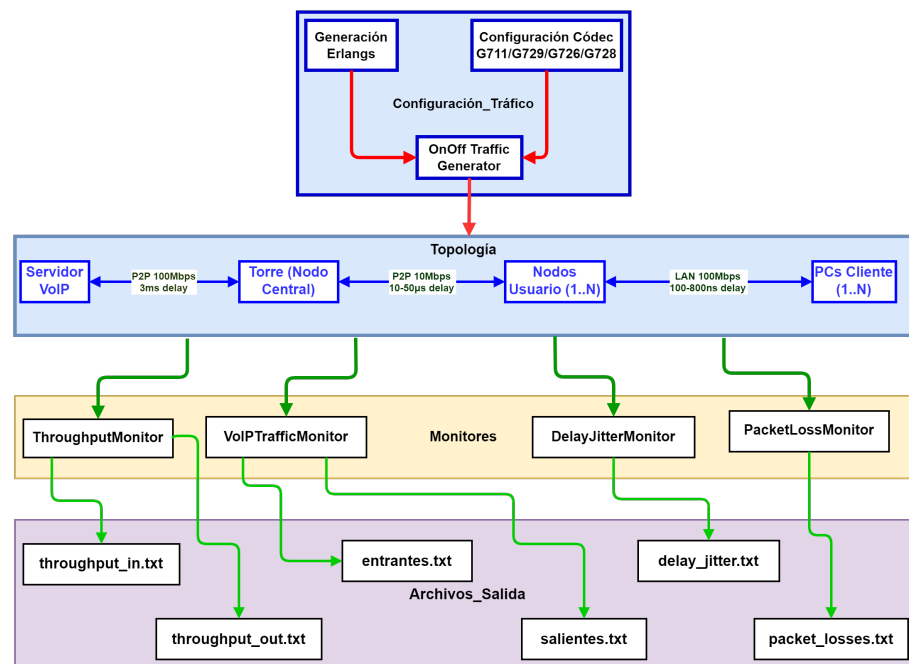
Los códecs G726a y G726b son opciones intermedias. El G726a (32 Kbps) maneja payloads de 80 bytes cada 20ms, logrando una eficiencia del 60.61 % y una tasa efectiva de 52.8 Kbps. El G726b (24 Kbps) usa payloads de 60 bytes cada 20ms, con una eficiencia del 53.57 % y tasa efectiva de 44.8 Kbps.

El G728 es el más “económico” en términos de consumo de datos, usando solo 16 Kbps. Con payloads de 60 bytes cada 30ms, mantiene una eficiencia del 53.57 % y una tasa efectiva de 29.87 Kbps. Su intervalo más largo (30ms vs 20ms) resulta en menos paquetes por segundo (33.33 vs 50), lo que puede ayudar a reducir la carga de procesamiento en la red.

En términos de escalabilidad, para un enlace de 1 Mbps podríamos manejar aproximadamente 11 llamadas simultáneas con G711, 34 con G729, 18 con G726a, 22 con G726b y 33 con G728. Esto explica la popularidad de G729 en implementaciones reales: aunque su eficiencia es baja debido al overhead, su bajo consumo de ancho de banda permite manejar más llamadas simultáneas con una calidad muy aceptable.

Para monitorear el desempeño del sistema, implementamos varios monitores que nos permiten medir métricas importantes como el throughput (cuántos datos están pasando por la red), el jitter (variación en los retardos), y las pérdidas de paquetes. Todos estos datos se guardan en archivos de texto para su posterior análisis.

Un aspecto crucial que simulamos es el retardo en diferentes partes de la red. Entre la PC del usuario y su router local (simulado con CSMA) el retardo es mínimo - estamos hablando de microsegundos. La conexión entre la torre y cada usuario (el famoso “última milla”) tiene un retardo variable entre 10 y 50 microsegundos. Finalmente, entre la torre y el servidor central tenemos un retardo de 3 ms, que es bastante realista para una red

Figura 4.4*Bloques de Simulador*

metropolitana.

Este simulador nos permite entender mejor cómo funcionaría una implementación real de VoIP y cómo diferentes decisiones técnicas afectan el rendimiento general del sistema. La elección del códec no solo depende de la calidad de audio deseada (MOS), sino también de consideraciones prácticas como la eficiencia del protocolo, el ancho de banda disponible y la cantidad de usuarios simultáneos que necesitamos soportar.

4.3.1. Simulación de Red de Acceso de Localidad de Huacaspata - Trafico VoIP

Según el cálculo de demanda tenemos en Huacaspata 24 llamadas entrantes y 28 llamadas salientes las que ha sido configuradas en el simulador de forma diferente e independiente para las llamadas entrantes y salientes para ajustarse mejor a la realidad

4.3.1.1. Trafico Entrante: Generado y en la Red

La Figura [4.5a](#) es el tráfico generado entrante:

- La gráfica muestra el tráfico bruto generado, donde se puede observar el impacto directo del tamaño de payload de cada códec:
 - G711 (azul): Con payload de 160 bytes, muestra los picos más altos (hasta 1600 Kbps).
 - G726a (naranja) y G726b (verde): Con payloads de 80 y 60 bytes respectivamente, muestran valores intermedios consistentes.
 - G728 (rojo): Con payload de 60 bytes cada 30ms, muestra valores bajos. Su menor tasa de paquetes por segundo (33.33 pps vs 50 pps) contribuyen a un tráfico más reducido.
 - G729 (morado): Con 20 bytes de payload, muestra los valores más bajos.

La Figura 4.5b es el tráfico medido en la red:

- G711: Mantiene un throughput normalizado 1.2Mbps, reflejando su alta tasa de bits y mejor eficiencia.
- G726a/b: Sus throughputs entre 0.6-0.8Mbps muestran el balance entre eficiencia moderada y tasas intermedias.
- G728 (rojo): Su throughput más bajo (0.25 Mbps) refleja tanto su eficiencia del 53.57% como su menor tasa de paquetes por segundo.
- G729 (morado): Similar al G728 en throughput (0.2-0.3Mbps), pero logrado a través de una menor carga útil con más paquetes por segundo.

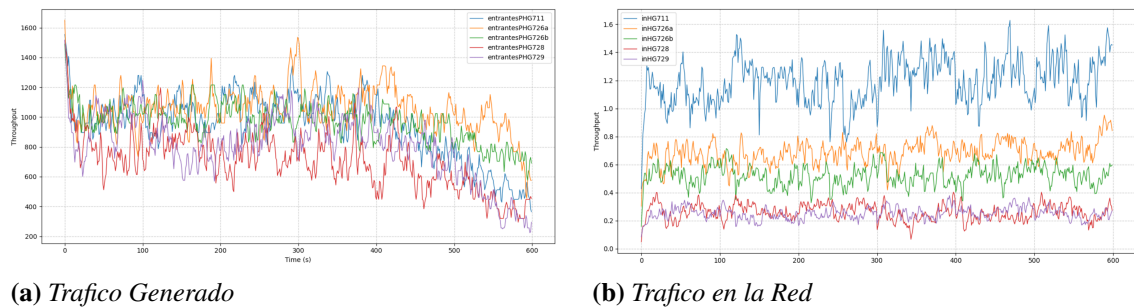
Análisis de Eficiencia y Características Distintivas:

1. Comparación G728 vs G729:

- Aunque ambos muestran throughput similar en la red, lo logran de manera diferente:
 - G728: Mayor eficiencia (53.57%) pero menos paquetes por segundo (33.33 pps)

Figura 4.5

Trafico de 28 Llamadas y Diferentes Codecs

**(a) Trafico Generado****(b) Trafico en la Red**

- G729: Menor eficiencia (27.78%) pero más paquetes por segundo (50 pps)

2. Impacto del Intervalo de Muestreo:

- G728 es único con su intervalo de 30ms:
 - Reduce la sobrecarga total al enviar menos paquetes por segundo
 - Compensa su tamaño de payload moderado (60 bytes) con menor frecuencia de transmisión
 - Esto resulta en un comportamiento más estable en la Figura [4.5b](#)

3. Eficiencia versus Frecuencia de Paquetes:

- La combinación de estos factores es importante:
 - G711: Alta eficiencia (75.47%) + alta frecuencia (50 pps) = mayor throughput
 - G728: Eficiencia media (53.57%) + baja frecuencia (33.33 pps) = throughput reducido
 - G729: Baja eficiencia (27.78%) + alta frecuencia (50 pps) = throughput similar a G728

4. Implicaciones para Huacaspata:

- Con 28 llamadas simultáneas:

- G728 y G729 ofrecen el menor consumo de ancho de banda por diferentes mecanismos
- G728 podría ofrecer ventajas en términos de procesamiento de red por su menor tasa de paquetes
- G729 mantiene una mayor frecuencia de actualización que podría beneficiar la calidad percibida

4.3.1.2. Trafico Entrante: Generado y en la Red

En la Figura [4.6a](#) tenemos el tráfico generado saliente

1. Comportamiento por Códec:

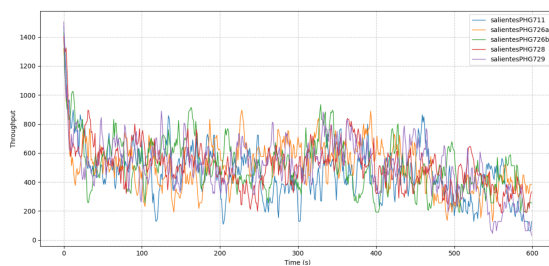
- G711 (azul):
 - Muestra valores más altos por su tasa de 64 Kbps
 - Payload puro de 160 bytes cada 20ms
 - Las fluctuaciones representan los patrones naturales de voz
- G726a (naranja):
 - Tráfico correspondiente a 32 Kbps
 - Payload puro de 80 bytes cada 20ms
 - Valores intermedios consistentes con su tasa de bits
- G726b (verde):
 - Refleja su tasa de 24 Kbps
 - Payload puro de 60 bytes cada 20ms
 - Comportamiento proporcional a su tasa de codificación
- G728 (rojo):
 - Representa 16 Kbps
 - Payload de 60 bytes cada 30ms
 - Menor frecuencia de paquetes por el intervalo más largo

- G729 (morado):
 - El más ligero con 8 Kbps
 - Payload puro de 20 bytes cada 20ms
 - Valores más bajos reflejando su alta compresión

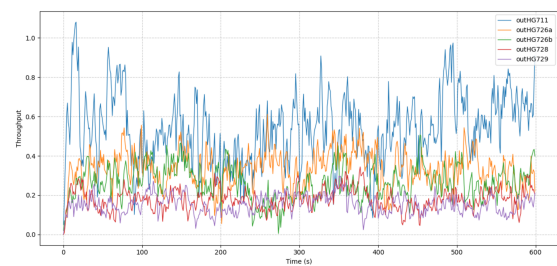
La Figura 4.6b se ve el tráfico en la red que esta afectado por la sobrecarga, por codec:

Figura 4.6

Trafico de 24 Llamadas y Diferentes Codecs



(a) Trafico Generado



(b) Trafico en la Red

1. G711

- Tráfico original: 160 bytes de voz
- Con sobrecarga: $160 + 52 = 212$ bytes (eficiencia 75.47%)
- Mayor throughput en la red pero la sobrecarga es proporcionalmente menor
- La sobrecarga representa 24.53% del paquete total

2. G726a

- Con sobrecarga: $80 + 52 = 132$ bytes (eficiencia 60.61%)
- La sobrecarga impacta más notoriamente que en G711
- Throughput intermedio con mayor proporción de overhead

3. G726b

- Con sobrecarga: $60 + 52 = 112$ bytes (eficiencia 53.57%)

- El impacto del overhead es más significativo
- La sobrecarga representa casi la mitad del paquete

4. G728

- Con sobrecarga: $60 + 52 = 112$ bytes (eficiencia 53.57%)
- Menor tasa de paquetes ayuda a reducir el impacto total de la sobrecarga
- Comportamiento más estable por su intervalo más largo

5. G729

- Con sobrecarga: $20 + 52 = 72$ bytes (eficiencia 27.78%)
- La sobrecarga domina el tamaño del paquete (72.22)
- El throughput más bajo en datos puros se ve significativamente afectado por el overhead

Análisis para Huacaspata (24 llamadas):

1. Impacto de la Sobrecarga:

- La diferencia entre ambas gráficas muestra claramente cómo el overhead afecta más a los códecs de menor payload
- G729 sufre el mayor impacto proporcional, aunque mantiene el menor consumo total
- G711 muestra la mejor eficiencia pero el mayor consumo absoluto

2. Recomendaciones:

- A pesar de la baja eficiencia de protocolo de G729 y G728, siguen siendo opciones atractivas por su bajo consumo total
- La elección dependerá del balance entre calidad de voz requerida y recursos de red disponibles

- Para 24 llamadas, los códecs de menor tasa siguen siendo ventajosos a pesar de su menor eficiencia de protocolo

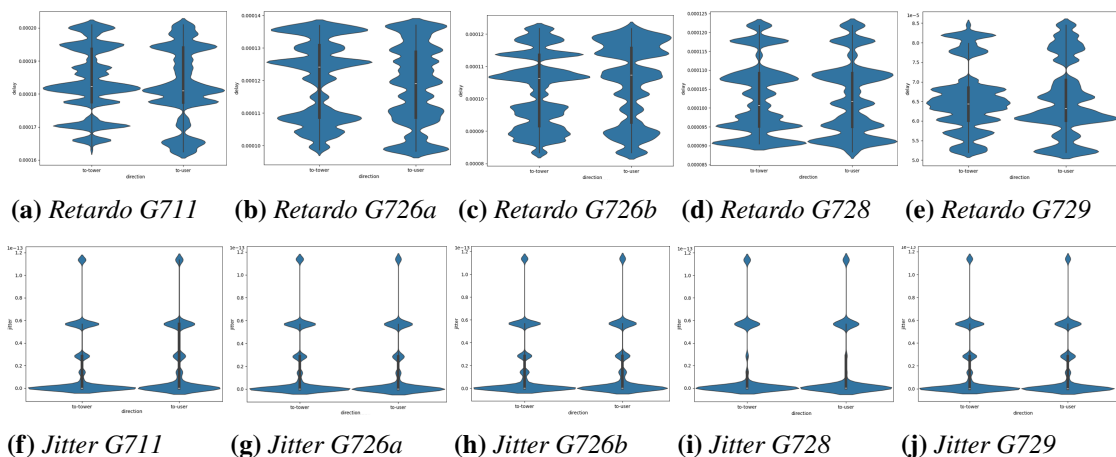
4.3.1.3. Retardo y Jitter

En términos de delay o retardo, se observa una progresión interesante en la eficiencia de los códecs. El G.729(Ver Figura 4.7e) se posiciona como el más eficiente, con valores entre 0.05 y 0.085 milisegundos, seguido por el G.726b(Figura 4.7c) (0.08-0.12 ms), el G.726a(Figura 4.7b) (0.10-0.14 ms), el G.728(Figura 4.7d) (0.85-0.125 ms) y finalmente el G.711 (0.16-0.20 ms). Esta progresión muestra una clara evolución en la optimización del retardo, donde cada generación de códec ha logrado mejoras significativas sobre su predecesor.

Un hallazgo particularmente interesante se relaciona con el jitter o variación en el tiempo de llegada de los paquetes. Sorprendentemente, todos los códecs muestran patrones prácticamente idénticos, con valores principalmente entre 0 y $0,6 \times 10^{-13}$ segundos y picos máximos que alcanzan los $1,2 \times 10^{-13}$ segundos. Esta notable consistencia en el jitter sugiere que este parámetro está más determinado por las características de la infraestructura de red que por el tipo de códec utilizado.

Figura 4.7

Retardo y Jitter de Localidad de Huacaspata



La estabilidad en las mediciones para ambas direcciones ("to-tower" "to-user") es

remarcable en todos los códecs, indicando una red bien configurada y equilibrada. Esto es especialmente relevante considerando que la localidad maneja 24 usuarios hacia la torre y 29 desde ella, sin que esta asimetría afecte significativamente el rendimiento.

Aunque el G.729 es la opción técnicamente superior en términos de delay, es importante notar que las diferencias entre los códecs se miden en microsegundos. En la práctica, cualquiera de estos códecs proporcionaría una calidad de servicio más que adecuada para comunicaciones de voz en Hucaspatha. La elección final podría basarse en otros factores como el consumo de ancho de banda, la compatibilidad con equipos existentes o consideraciones de costo, dado que todos muestran un rendimiento técnico aceptable.

La consistencia en el comportamiento del jitter a través de todos los códecs también sugiere que las mejoras futuras en la calidad del servicio podrían enfocarse más en optimizaciones de red que en la selección del códec, ya que la variabilidad en los tiempos de llegada parece estar más influenciada por la infraestructura que por el algoritmo de codificación utilizado.

El análisis comparativo de los parámetros de calidad de servicio (retardo y jitter) a través de diferentes localidades y códecs revela patrones interesantes y contrastes significativos en el comportamiento de la red.

- Comportamiento del Retardo: En la Figura 4.8 tenemos a los códecs que muestran una jerarquía clara y consistente en términos de retardo. El G729 destaca como el más eficiente, manteniendo valores por debajo de 0.075 ms en todas las localidades. Le siguen en orden de eficiencia el G728 (0.100 ms), G726B (0.095-0.110 ms), G726A (0.115-0.125 ms), y finalmente el G711, que consistentemente muestra los valores más altos (0.175 ms). Esta progresión se mantiene relativamente estable a través de todas las localidades, sugiriendo que las características intrínsecas de los códecs son más determinantes que las variables geográficas.
- Comportamiento del Jitter: En la Figura 4.9, el jitter presenta un panorama más complejo y variable. Los valores, aunque extremadamente bajos (orden de $10^{-11}ms$), muestran patrones distintivos por localidad:

- Coyartuna exhibe los valores más altos, particularmente con G726B ($2,3 \times 10^{-11}ms$)
 - Chocobamba muestra picos con G726A y G729 ($2,1 \times 10^{-11}ms$)
 - Fortaleza Andina presenta la distribución más uniforme ($1,8 \times 10^{-11}ms$)
- Análisis por Localidades:
1. Fortaleza Andina y Chocobamba:
 - Muestran los retardos más bajos consistentemente
 - Fortaleza Andina presenta mejor estabilidad en jitter
 - Chocobamba muestra variabilidad significativa en jitter con ciertos códecs
 2. Coyartuna y Buenos Aires:
 - Presentan los retardos más elevados
 - Coyartuna muestra la mayor variabilidad en jitter
 - Buenos Aires mantiene valores intermedios de jitter
 3. Huacaspata y Extremo:
 - Huacaspata destaca por retardos inferiores al promedio
 - Extremo mantiene valores intermedios de retardo
 - Ambas muestran patrones distintivos de jitter, con picos específicos por códec

Figura 4.8

Retardo Promedio por Localidad y codec

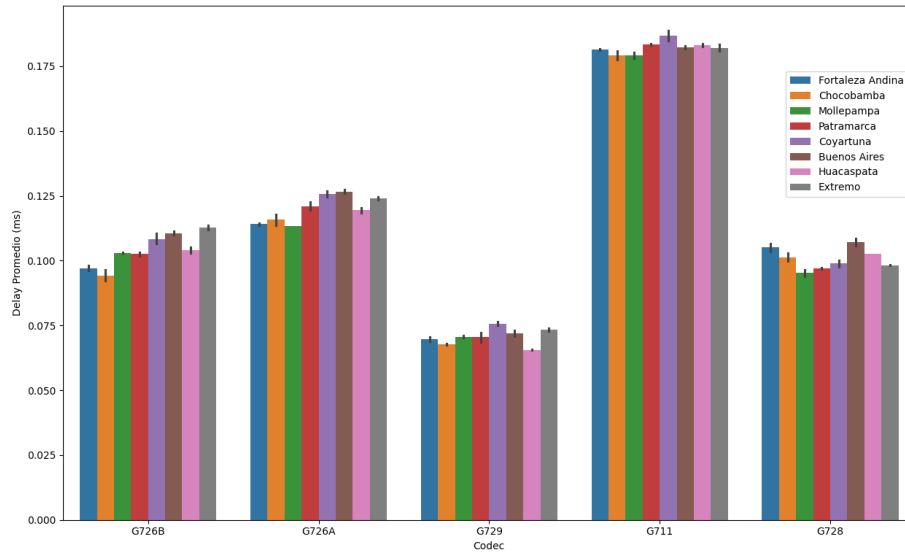
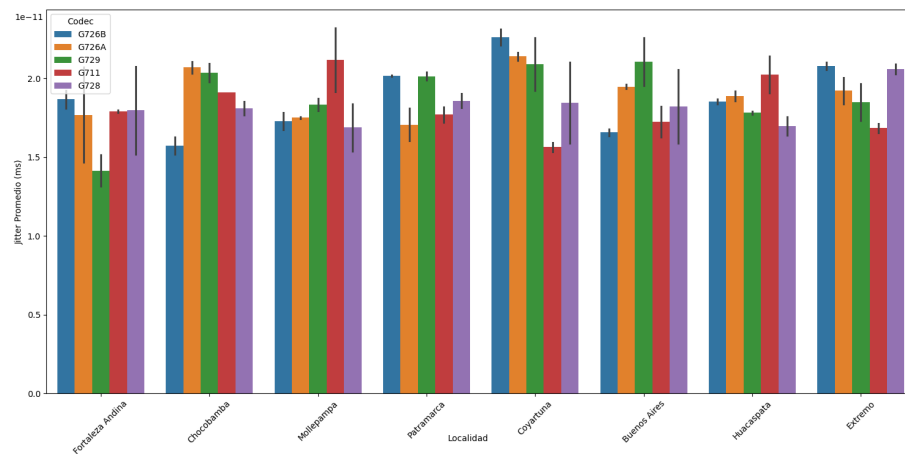


Figura 4.9

Jitter Promedio por Localidad y codec



Conclusiones

El diagnóstico y estimación de demanda realizado reveló un patrón significativo de necesidades de conectividad en las 13 localidades beneficiadas. Los análisis demostraron que estas comunidades requieren en promedio entre 24 y 28 llamadas simultáneas, con una demanda agregada que oscila entre 0.2 y 1.2 Mbps dependiendo del códec utilizado, evidenciando la importancia de una planificación cuidadosa de recursos.

En cuanto al impacto de diferentes códecs en el ancho de banda, los resultados fueron reveladores. El G.711, aunque ofrece la mejor calidad de audio con un MOS de 4.1, requiere significativamente más ancho de banda (64 Kbps), mientras que el G.729 logra una excelente eficiencia utilizando solo 8 Kbps con una calidad aceptable (MOS 3.92). Esta diferencia tiene implicaciones directas en la capacidad de la red para manejar llamadas simultáneas, donde una conexión de 1 Mbps podría soportar 11 llamadas con G.711 o hasta 34 llamadas con G.729.

El análisis de métricas clave mostró resultados interesantes en términos de latencia y jitter. El G.729 demostró ser el más eficiente con retardos entre 0.05 y 0.085 milisegundos, seguido por el G.726b (0.08-0.12 ms) y el G.711 (0.16-0.20 ms). Sorprendentemente, el jitter se mantuvo consistente entre todos los códecs, oscilando entre 0 y $0,610^{-13}$ segundos, sugiriendo que esta métrica está más influenciada por la infraestructura de red que por la elección del códec.

Finalmente, integrando todos estos hallazgos, podemos concluir que la elección del códec tiene un impacto significativo en la asignación de recursos de la red inalámbrica de banda ancha en Huacaspata. El G.729 emerge como una solución particularmente atractiva para este entorno rural, ofreciendo un equilibrio óptimo entre calidad de servicio y eficiencia en el uso de recursos. No obstante, todos los códecs evaluados demostraron ser técnicamente viables, manteniendo parámetros de QoS dentro de rangos aceptables, lo que sugiere que la selección final puede basarse en consideraciones adicionales como la disponibilidad de equipos y costos de implementación.

Bibliografía

- Adhilaksono, B., & Setiawan, B. (2022). A Study of Voice-over-Internet Protocol Quality Metrics. *Procedia Computer Science*, 197, 377-384. <https://doi.org/10.1016/j.procs.2021.12.153>
- Burbur, H. M., Abdalla, K. H. B., Elhassan, A. M., Elhassan, F. D. M., Osman, T. W., & Khider, I. (2022). VOIP QoS Over UMTS Network Using IEEE 802.11e EDCA Algorithm. *2022 International Conference on Smart Information Systems and Technologies (SIST)*, 1-7. <https://doi.org/10.1109/SIST54437.2022.9945805>
- Estela Perez, A., & Jimenez Requejo, S. S. (2024). *Análisis de desempeño de red de banda ancha rural - caso de estudio distrito de Shamboyacu, San Martin* [Tesis de Grado]. Universidad Nacional Pedro Ruiz Gallo. Consultado el 2 de diciembre de 2024, desde <http://repositorio.unprg.edu.pe/handle/20.500.12893/12782>
Accepted: 2024-03-25T21:58:15Z.
- INEI. (2018). INEI - REDATAM CENSOS 2017. *INEI*. <https://censos2017.inei.gob.pe/redatam/>
- Manap, Z., Mohd Zain, A. S., Mohd Said, R., & Shivaneson Balakirisnan, S. (2020). Analysis of 4G Mobile Network Coverage in UTeM Technology Campus. *Indonesian Journal of Electrical Engineering and Computer Science*, 18(2), 882. <https://doi.org/10.11591/ijeecs.v18.i2.pp882-889>
- Maslen, A., & Nunez, M. (2021). *Customized NS-3 Simulation for Bandwidth Assessment of Rural Communities* [Tesis de Grado]. Worcester Polytechnic Institute. [maslenaCustomizedNS3Simulation2021](https://doi.org/10.1109/NS3Simulation2021).
- Mauro, M. D., & Liotta, A. (2020). An Experimental Evaluation and Characterization of VoIP over an LTE-A Network. *IEEE Transactions on Network and Service Management*, 17(3), 1626-1639. <https://doi.org/10.1109/TNSM.2020.2995505>
- Mauro, M., Galatro, G., Postiglione, F., Song, W., & Liotta, A. (2023). Multivariate Time Series Characterization and Forecasting of VoIP Traffic in Real Mobile Networks. *IEEE Transactions on Network and Service Management*, 1-1. <https://doi.org/10.1109/TNSM.2023.3295748>

- Nedyalkov, I. (2023). Benefits of Using Network Modeling Platforms When Studying IP Networks and Traffic Characterization. *Computers*, 12(2), 41. <https://doi.org/10.3390/computers12020041>
- Olorunnisola, O., Mathonsi, T. E., & Plessis, D. D. (2023). An Algorithm to Optimize Concurrent VoIP Calls Across Wireless Mesh Networks. *Journal of Advances in Information Technology*, 14(5), 960-969. <https://doi.org/10.12720/jait.14.5.960-969>
- Strzeciwilk, D. (2021). Performance Analysis of VoIP Data over IP Networks. *International Journal of Electronics and Telecommunications*, 67(4), 743-750. <https://doi.org/10.24425/ijet.2021.139801>

Carpeta compartida y programa usado

Programa escenarios VoIp Rural

```

1  #include "ns3/core-module.h"
2  #include "ns3/network-module.h"
3  #include "ns3/internet-module.h"
4  #include "ns3/point-to-point-module.h"
5  #include "ns3/csma-module.h"
6  #include "ns3/applications-module.h" // Asegurar que esté presente
7  #include <fstream>
8  #include <vector>
9  #include <sstream>
10 #include <iomanip>
11
12 using namespace ns3;
13
14 // Configuración global
15 uint32_t numUsuarios = 28; // Ejemplo
16 uint32_t Nser = 24; // Cantidad de PCs que reciben respuesta del servidor
17 std::string codec = "G711"; // Códec por defecto definido globalmente
18 Time simulationTime = Seconds(600.0);
19 std::string dataRateServerToTower = "100Mbps";
20 std::string delayServerToTower = "3ms";
21 std::string dataRateLAN = "100Mbps";
22
23
24
25 // Función auxiliar que dado un códec, retorna (DataRate, PacketSize)
26 std::pair<std::string, uint32_t> GetCodecParameters(const std::string &codecName) {
27     // codec    vtx(Kbps)    tam(bytes)
28     // G711     64           160
29     // G729     8            20
30     // G726a    32           80
31     // G726b    24           60
32     // G728     16           60
33
34     if (codecName == "G729") {
35         return {"8kbps", 20};
36     } else if (codecName == "G726a") {
37         return {"32kbps", 80};
38     } else if (codecName == "G726b") {
39         return {"24kbps", 60};
40     } else if (codecName == "G728") {
41         return {"16kbps", 60};
42     } else {
43         // Por defecto G711
44         return {"64kbps", 160};
45     }
46 }
47
48
49 // Generar valores de Erlang
50 std::vector<double> GenerateErlangs(uint32_t numValues, double mean, double stddev) {
51     std::vector<double> erlangs;
52     Ptr<NormalRandomVariable> rng = CreateObject<NormalRandomVariable>();
53     rng->SetAttribute("Mean", DoubleValue(mean));
54     rng->SetAttribute("Variance", DoubleValue(stddev * stddev));
55
56     for (uint32_t i = 0; i < numValues; ++i) {
57         double value = rng->GetValue();
58         erlangs.push_back(std::max(0.0, std::min(1.0, value)));
59     }
60     return erlangs;
61 }
62
63 void ConfigureErrorModel(Ptr<NetDevice> device, double errorRate) {
64     Ptr<RateErrorModel> errorModel = CreateObject<RateErrorModel>();
65     errorModel->SetAttribute("ErrorRate", DoubleValue(errorRate));
66     device->SetAttribute("ReceiveErrorModel", PointerValue(errorModel));
67 }
68
69 class ThroughputMonitor : public Object {
70 public:
71     void Initialize(const std::string &filename) {
72         m_outFile.open(filename);
73         if (!m_outFile.is_open()) {

```

```

74         NS_FATAL_ERROR("No se pudo abrir el archivo " << filename);
75     }
76     m_outFile << "time th\n";
77 }
78
79 void PacketEvent(std::string context, Ptr<const Packet> packet) {
80     m_bytes += packet->GetSize();
81     if (Simulator::Now().GetSeconds() >= m_nextTime) {
82         RecordThroughput(Simulator::Now().GetSeconds(), m_bytes);
83         m_bytes = 0;
84         m_nextTime += m_windowSize;
85     }
86 }
87
88 private:
89     void RecordThroughput(double time, uint64_t bytes) {
90         double throughput = (bytes * 8.0) / (m_windowSize * 1000000); // Mbps
91         m_outFile << time << " " << throughput << "\n";
92         m_outFile.flush();
93     }
94
95     std::ofstream m_outFile;
96     double m_windowSize = 0.3;
97     uint64_t m_bytes = 0;
98     double m_nextTime = 0.0;
99 };
100
101 class VoIPTrafficMonitor : public Object {
102 public:
103     void Initialize(const std::string &filename) {
104         m_outFile.open(filename);
105         if (!m_outFile.is_open()) {
106             NS_FATAL_ERROR("No se pudo abrir el archivo " << filename);
107         }
108         m_outFile << "userId onTime offTime dataRate\n";
109     }
110
111     void RegisterCall(uint32_t userId, double onTime, double offTime, double dataRate)
112     {
113         m_outFile << userId << " " << onTime << " " << offTime << " " << dataRate <<
114             "\n";
115         m_outFile.flush();
116     }
117 private:
118     std::ofstream m_outFile;
119 };
120
121 class DelayJitterMonitor : public Object {
122 public:
123     void Initialize(const std::string &filename) {
124         m_outFile.open(filename);
125         if (!m_outFile.is_open()) {
126             NS_FATAL_ERROR("No se pudo abrir el archivo " << filename);
127         }
128         m_outFile << "time userId direction delay jitter\n";
129     }
130
131     void PacketSent(uint32_t userId, const std::string &direction, Ptr<const Packet>
132     packet) {
133         uint64_t packetId = packet->GetUId();
134         double timeSent = Simulator::Now().GetSeconds();
135         m_packetSentTimes[packetId] = {timeSent, direction};
136         m_packetUserIds[packetId] = userId;
137     }
138
139     void PacketReceived(uint32_t userId, const std::string &direction, Ptr<const
140     Packet> packet) {
141         uint64_t packetId = packet->GetUId();
142         double timeReceived = Simulator::Now().GetSeconds();
143
144         if (m_packetSentTimes.find(packetId) != m_packetSentTimes.end()) {
145             double timeSent = m_packetSentTimes[packetId].first;

```

```

143         std::string sentDirection = m_packetSentTimes[packetId].second;
144
145         if (sentDirection == direction) {
146             double delay = timeReceived - timeSent;
147             double jitter = 0.0;
148             std::string key = std::to_string(userId) + "-" + direction;
149             if (m_lastDelays.find(key) != m_lastDelays.end()) {
150                 jitter = std::abs(delay - m_lastDelays[key]);
151             }
152             m_lastDelays[key] = delay;
153             RecordDelayJitter(Simulator::Now().GetSeconds(), userId, direction,
154                             delay, jitter);
155             m_packetSentTimes.erase(packetId);
156             m_packetUserIds.erase(packetId);
157         }
158     }
159
160     static void HandlePacketSent(Ptr<DelayJitterMonitor> monitor, uint32_t userId,
161     const std::string &direction, std::string context, Ptr<const Packet> packet) {
162         monitor->PacketSent(userId, direction, packet);
163     }
164
165     static void HandlePacketReceived(Ptr<DelayJitterMonitor> monitor, uint32_t userId,
166     const std::string &direction, std::string context, Ptr<const Packet> packet) {
167         monitor->PacketReceived(userId, direction, packet);
168     }
169 private:
170     void RecordDelayJitter(double time, uint32_t userId, const std::string &direction,
171     double delay, double jitter) {
172         m_outFile << time << " " << userId << " " << direction << " " << delay << " "
173         << jitter << "\n";
174         m_outFile.flush();
175     }
176
177     std::ofstream m_outFile;
178     std::map<uint64_t, std::pair<double, std::string>> m_packetSentTimes;
179     std::map<uint64_t, uint32_t> m_packetUserIds;
180     std::map<std::string, double> m_lastDelays;
181 };
182
183 class PacketLossMonitor : public Object {
184 public:
185     void Initialize(const std::string &filename) {
186         m_outFile.open(filename);
187         if (!m_outFile.is_open()) {
188             NS_FATAL_ERROR("No se pudo abrir el archivo " << filename);
189         }
190         m_outFile << "time userId losses\n";
191     }
192
193     void PacketDropped(uint32_t userId, std::string context, Ptr<const Packet> packet)
194     {
195         m_lossCounts[userId]++;
196         RecordLoss(Simulator::Now().GetSeconds(), userId);
197     }
198
199     static void HandlePacketDrop(Ptr<PacketLossMonitor> monitor, uint32_t userId, std
200     ::string context, Ptr<const Packet> packet) {
201         monitor->PacketDropped(userId, context, packet);
202     }
203 private:
204     void RecordLoss(double time, uint32_t userId) {
205         m_outFile << time << " " << userId << " " << m_lossCounts[userId] << "\n";
206         m_outFile.flush();
207     }
208
209     std::ofstream m_outFile;
210     std::map<uint32_t, uint32_t> m_lossCounts;
211 };

```

```

209 void ConfigureVoIPTraffic(uint32_t userId, Ipv4Address pcAddress, uint16_t port,
210                          double erlangToServer, double erlangFromServer,
211                          NodeContainer pcNode, NodeContainer serverNode,
212                          Ipv4InterfaceContainer serverInterfaces, Time stopTime,
213                          Ptr<VoIPTrafficMonitor> inMonitor, Ptr<VoIPTrafficMonitor>
214                          outMonitor) {
215     Ptr<UniformRandomVariable> sessionCountRng = CreateObject<UniformRandomVariable>
216     >();
217     Ptr<ExponentialRandomVariable> onTimeRngToServer = CreateObject<
218     ExponentialRandomVariable>();
219     Ptr<ExponentialRandomVariable> offTimeRngToServer = CreateObject<
220     ExponentialRandomVariable>();
221     Ptr<ExponentialRandomVariable> onTimeRngFromServer = CreateObject<
222     ExponentialRandomVariable>();
223     Ptr<ExponentialRandomVariable> offTimeRngFromServer = CreateObject<
224     ExponentialRandomVariable>();
225
226     uint32_t numSessions = sessionCountRng->GetInteger(15, 30);
227     double sessionTime = stopTime.GetSeconds() / numSessions;
228
229     onTimeRngToServer->SetAttribute("Mean", DoubleValue(erlangToServer * sessionTime
230     ));
231     offTimeRngToServer->SetAttribute("Mean", DoubleValue((1 - erlangToServer) *
232     sessionTime));
233
234     if (erlangFromServer > 0.0) {
235         onTimeRngFromServer->SetAttribute("Mean", DoubleValue(erlangFromServer *
236         sessionTime));
237         offTimeRngFromServer->SetAttribute("Mean", DoubleValue((1 - erlangFromServer)
238         * sessionTime));
239     }
240     // Obtener parámetros del códec elegido
241     std::pair<std::string, uint32_t> codecParams = GetCodecParameters(codec);
242     std::string dataRate = codecParams.first;
243     uint32_t packetSize = codecParams.second;
244
245     // Crear PacketSink en el servidor para recibir tráfico desde la PC
246     {
247         PacketSinkHelper sink("ns3::UdpSocketFactory", InetSocketAddress(Ipv4Address::
248         GetAny(), port));
249         ApplicationContainer sinkApp = sink.Install(serverNode.Get(0));
250         sinkApp.Start(Seconds(0.0));
251         sinkApp.Stop(stopTime);
252     }
253
254     // Tráfico PC->Servidor (OnOff en la PC)
255     for (uint32_t session = 0; session < numSessions; ++session) {
256         double onTimeToServer = onTimeRngToServer->GetValue();
257         double offTimeToServer = offTimeRngToServer->GetValue();
258
259         OnOffHelper onOffToServer("ns3::UdpSocketFactory", InetSocketAddress(
260         serverInterfaces.GetAddress(0), port));
261         onOffToServer.SetAttribute("DataRate", StringValue(dataRate));
262         onOffToServer.SetAttribute("PacketSize", UIntegerValue(packetSize));
263         onOffToServer.SetAttribute("OnTime", StringValue(
264         "ns3::ConstantRandomVariable[Constant=" + std::to_string(onTimeToServer) + "]"
265         ));
266         onOffToServer.SetAttribute("OffTime", StringValue(
267         "ns3::ConstantRandomVariable[Constant=" + std::to_string(offTimeToServer) +
268         "]"));
269         ApplicationContainer toServerApp = onOffToServer.Install(pcNode.Get(0));
270         toServerApp.Start(Seconds(session * sessionTime));
271         toServerApp.Stop(Seconds((session + 1) * sessionTime));
272
273         if (inMonitor) {
274             double dataRateKbps = std::stod(dataRate.substr(0, dataRate.size() - 4));
275             // extrae el valor de XXkbps
276             inMonitor->RegisterCall(userId, onTimeToServer, offTimeToServer,
277             dataRateKbps * 1000);
278         }
279     }
280 }

```

```

264 // Si el servidor va a enviar tráfico a la PC
265 if (erlangFromServer > 0.0) {
266     // PacketSink en la PC para recibir tráfico desde el servidor
267     {
268         PacketSinkHelper sinkPC("ns3::UdpSocketFactory", InetSocketAddress(
269             Ipv4Address::GetAny(), port + 10000));
270         ApplicationContainer sinkPCApp = sinkPC.Install(pcNode.Get(0));
271         sinkPCApp.Start(Seconds(0.0));
272         sinkPCApp.Stop(stopTime);
273     }
274     for (uint32_t session = 0; session < numSessions; ++session) {
275         double onTimeFromServer = onTimeRngFromServer->GetValue();
276         double offTimeFromServer = offTimeRngFromServer->GetValue();
277
278         OnOffHelper onOffFromServer("ns3::UdpSocketFactory", InetSocketAddress(
279             pcAddress, port + 10000));
280         onOffFromServer.SetAttribute("DataRate", StringValue(dataRate));
281         onOffFromServer.SetAttribute("PacketSize", UIntegerValue(packetSize));
282         onOffFromServer.SetAttribute("OnTime", StringValue(
283             "ns3::ConstantRandomVariable[Constant=" + std::to_string(onTimeFromServer)
284             + "]"));
285         onOffFromServer.SetAttribute("OffTime", StringValue(
286             "ns3::ConstantRandomVariable[Constant=" + std::to_string(offTimeFromServer)
287             + "]"));
288         ApplicationContainer fromServerApp = onOffFromServer.Install(serverNode.
289             Get(0));
290         fromServerApp.Start(Seconds(session * sessionTime));
291         fromServerApp.Stop(Seconds((session + 1) * sessionTime));
292     }
293     if (outMonitor) {
294         double dataRateKbps = std::stod(dataRate.substr(0, dataRate.size() - 4
295             ));
296         outMonitor->RegisterCall(userId, onTimeFromServer, offTimeFromServer,
297             dataRateKbps * 1000);
298     }
299 }
300
301 int main(int argc, char *argv[]) {
302     CommandLine cmd;
303     cmd.AddValue("numUsuarios", "Número de usuarios conectados", numUsuarios);
304     cmd.AddValue("Nser", "Número de usuarios que reciben respuesta del servidor", Nser);
305     cmd.AddValue("codec", "Tipo de codec a utilizar (G711, G729, G726a, G726b, G728)",
306         codec);
307     cmd.Parse(argc, argv);
308
309     time_t now = time(0);
310     RngSeedManager::SetSeed (now);
311     double delayThreshold = 40.0;
312     double errorRate = 0.0001;
313
314     auto erlangstoServer = GenerateErlangs(numUsuarios, 0.55, 0.4);
315     auto erlangstoFromServer = GenerateErlangs(numUsuarios, 0.35, 0.1);
316
317     NodeContainer serverNode, torreNode, userNodes, pcNodes;
318     serverNode.Create(1);
319     torreNode.Create(1);
320     userNodes.Create(numUsuarios);
321     pcNodes.Create(numUsuarios);
322
323     PointToPointHelper p2pServerToTower;
324     p2pServerToTower.SetDeviceAttribute("DataRate", StringValue(dataRateServerToTower
325         ));
326     p2pServerToTower.SetChannelAttribute("Delay", StringValue(delayServerToTower));
327     NetDeviceContainer serverTorreDevices = p2pServerToTower.Install(serverNode.Get(0
328         ), torreNode.Get(0));
329
330     std::vector<NetDeviceContainer> torreUserDevices;
331     for (uint32_t i = 0; i < numUsuarios; ++i) {
332         PointToPointHelper p2pTowerToUser;

```

```

324     p2pTowerToUser.SetDeviceAttribute("DataRate", StringValue("10Mbps"));
325     Ptr<UniformRandomVariable> delayRng = CreateObject<UniformRandomVariable>(
326     double delayValue = delayRng->GetValue(10, 50);
327     std::ostringstream delayString;
328     delayString << delayValue << "us";
329     p2pTowerToUser.SetChannelAttribute("Delay", StringValue(delayString.str()));
330     NetDeviceContainer link = p2pTowerToUser.Install(torreNode.Get(0), userNode
Get(i));
331     torreUserDevices.push_back(link);
332     if (delayValue > delayThreshold) {
333         ConfigureErrorModel(link.Get(1), errorRate);
334     }
335 }
336
337 CsmHelper csma;
338 csma.SetChannelAttribute("DataRate", StringValue(dataRateLAN));
339 std::vector<NetDeviceContainer> userPCDevices;
340 for (uint32_t i = 0; i < numUsuarios; ++i) {
341     Ptr<UniformRandomVariable> delayRng = CreateObject<UniformRandomVariable>(
342     double delayValue = delayRng->GetValue(100, 800);
343     std::ostringstream delayString;
344     delayString << delayValue << "ns";
345     csma.SetChannelAttribute("Delay", StringValue(delayString.str()));
346     NetDeviceContainer lan = csma.Install(NodeContainer(userNodes.Get(i), pcNodes
Get(i)));
347     userPCDevices.push_back(lan);
348 }
349
350 InternetStackHelper stack;
351 stack.Install(serverNode);
352 stack.Install(torreNode);
353 stack.Install(userNodes);
354 stack.Install(pcNodes);
355
356 Ipv4AddressHelper address;
357 address.SetBase("10.1.1.0", "255.255.255.0");
358 Ipv4InterfaceContainer serverInterfaces = address.Assign(serverTorreDevices);
359
360 std::vector<Ipv4InterfaceContainer> userInterfaces, pcInterfaces;
361 for (uint32_t i = 0; i < numUsuarios; ++i) {
362     std::ostringstream subnet;
363     subnet << "10.2." << (i + 2) << ".0";
364     address.SetBase(subnet.str().c_str(), "255.255.255.0");
365     userInterfaces.push_back(address.Assign(torreUserDevices[i]));
366
367     std::ostringstream lanSubnet;
368     lanSubnet << "10.3." << (i + 10) << ".0";
369     address.SetBase(lanSubnet.str().c_str(), "255.255.255.0");
370     pcInterfaces.push_back(address.Assign(userPCDevices[i]));
371 }
372
373 Ptr<VoIPTrafficMonitor> inMonitor = CreateObject<VoIPTrafficMonitor>();
374 inMonitor->Initialize("entrantes.txt");
375
376 Ptr<VoIPTrafficMonitor> outMonitor = CreateObject<VoIPTrafficMonitor>();
377 outMonitor->Initialize("salientes.txt");
378
379 Ptr<ThroughputMonitor> throughputInMonitor = CreateObject<ThroughputMonitor>();
380 throughputInMonitor->Initialize("throughput_in.txt");
381
382 Ptr<ThroughputMonitor> throughputOutMonitor = CreateObject<ThroughputMonitor>();
383 throughputOutMonitor->Initialize("throughput_out.txt");
384
385 for (uint32_t i = 0; i < numUsuarios; ++i) {
386     double downErlang = (i < Nserver) ? erlangsFromServer[i] : 0.0;
387     ConfigureVoIPTraffic(i, pcInterfaces[i].GetAddress(1), 5060 + i,
erlangsToServer[i], downErlang, NodeContainer(pcNodes.Get(i)), serverNode,
serverInterfaces, simulationTime, inMonitor, outMonitor);
388 }
389
390 Ptr<DelayJitterMonitor> delayJitterMonitor = CreateObject<DelayJitterMonitor>();
391 delayJitterMonitor->Initialize("delay_jitter.txt");
392

```

```

393     for (uint32_t i = 0; i < numUsuarios; ++i) {
394         Ptr<NetDevice> torreDevice = torreUserDevices[i].Get(0);
395         Ptr<NetDevice> userDevice = torreUserDevices[i].Get(1);
396
397         std::string txPathToUser = "/NodeList/" + std::to_string(torreDevice->GetNode
398             (->GetId()) +
399             "/DeviceList/" + std::to_string(torreDevice->
400             GetIfIndex()) +
401             "/$ns3::PointToPointNetDevice/MacTx";
402         std::string rxPathToUser = "/NodeList/" + std::to_string(userDevice->GetNode
403             (->GetId()) +
404             "/DeviceList/" + std::to_string(userDevice->
405             GetIfIndex()) +
406             "/$ns3::PointToPointNetDevice/MacRx";
407
408         Config::Connect(txPathToUser, MakeBoundCallback(&DelayJitterMonitor::
409             HandlePacketSent, delayJitterMonitor, i, "to-user"));
410         Config::Connect(rxPathToUser, MakeBoundCallback(&DelayJitterMonitor::
411             HandlePacketReceived, delayJitterMonitor, i, "to-user"));
412
413         std::string txPathToTower = "/NodeList/" + std::to_string(userDevice->GetNode
414             (->GetId()) +
415             "/DeviceList/" + std::to_string(userDevice->
416             GetIfIndex()) +
417             "/$ns3::PointToPointNetDevice/MacTx";
418         std::string rxPathToTower = "/NodeList/" + std::to_string(torreDevice->GetNode
419             (->GetId()) +
420             "/DeviceList/" + std::to_string(torreDevice->
421             GetIfIndex()) +
422             "/$ns3::PointToPointNetDevice/MacRx";
423
424         Config::Connect(txPathToTower, MakeBoundCallback(&DelayJitterMonitor::
425             HandlePacketSent, delayJitterMonitor, i, "to-tower"));
426         Config::Connect(rxPathToTower, MakeBoundCallback(&DelayJitterMonitor::
427             HandlePacketReceived, delayJitterMonitor, i, "to-tower"));
428     }
429
430     Ptr<PacketLossMonitor> lossMonitor = CreateObject<PacketLossMonitor>();
431     lossMonitor->Initialize("packet_losses.txt");
432     for (uint32_t i = 0; i < numUsuarios; ++i) {
433         Ptr<NetDevice> userDevice = torreUserDevices[i].Get(1);
434         std::string path = "/NodeList/" + std::to_string(userDevice->GetNode(->GetId
435             ()) +
436             "/DeviceList/" + std::to_string(userDevice->GetIfIndex()) +
437             "/$ns3::PointToPointNetDevice/PhyRxDrop";
438         Config::Connect(path, MakeBoundCallback(&PacketLossMonitor::HandlePacketDrop,
439             lossMonitor, i));
440     }
441
442     Config::Connect("/NodeList/" + std::to_string(serverNode.Get(0)->GetId()) +
443         "/DeviceList/*/$ns3::PointToPointNetDevice/MacRx",
444         MakeCallback(&ThroughputMonitor::PacketEvent, throughputInMonitor
445             ));
446
447     Config::Connect("/NodeList/" + std::to_string(serverNode.Get(0)->GetId()) +
448         "/DeviceList/*/$ns3::PointToPointNetDevice/MacTx",
449         MakeCallback(&ThroughputMonitor::PacketEvent, throughputOutMonitor
450             ));
451
452     Ipv4GlobalRoutingHelper::PopulateRoutingTables();
453     Simulator::Stop(simulationTime);
454     Simulator::Run();
455     Simulator::Destroy();
456
457     return 0;
458 }
459

```