

**UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"**

**MODELO DE PREDICCIÓN DE CALIDAD DE SERVICIO PARA LA RED
DE TELEFONIA DIGITAL DE INTERCABLE**

NAYVIC LILIANA PIÑA FINZI

|
Barquisimeto, 2007

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIA Y TECNOLOGIA
MAESTRIA EN CIENCIAS DE LA COMPUTACION**

**MODELO DE PREDICCIÓN DE CALIDAD DE SERVICIO PARA LA RED
DE TELEFONIA DIGITAL DE INTERCABLE**

**Trabajo presentado para optar al Grado de
Magíster Scientiarum**

Por: NAYVIC LILIANA PIÑA FINZI

Barquisimeto, 2007

DEDICATORIA

A Dios Todopoderoso.

A mis Padres Victor y Nadia.

A mi hijo Manuel Alejandro.

A mi novio, compañero y amigo Niels.

A mis hermanos Victor, Nadya y Rita.

A mi sobrina Isabela.

AGRADECIMIENTO

A Dios Todopoderoso, porque con su guía alcancé tan importante meta.

A mis padres porque siempre son un ejemplo a seguir y por su apoyo en todo momento

A mi hijo Manuel Alejandro porque todos mis logros son y siempre serán para él.

A mi novio Niels por su apoyo, comprensión, paciencia y colaboración durante el cumplimiento de esta importante meta.

A mi tutor Prof. Arsenio Pérez por su apoyo prestado.

A todos los que de una u otra forma me ayudaron alcanzar este logro.

INDICE GENERAL

	pág.
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
INDICE.....	vi
INDICE DE TABLAS.....	viii
INDICE DE FIGURAS.....	ix
RESUMEN.....	x
INTRODUCCIÓN.....	1
CAPITULO	
I EL PROBLEMA	3
Planteamiento del Problema.....	3
Objetivos de la Investigación.....	4
General.....	4
Específicos.....	4
Justificación e Importancia.....	5
Alcances y Limitaciones.....	6
II MARCO TEORICO	8
Antecedentes.....	8
Bases Teóricas.....	11
Inicios de la VoIP	11
Componentes de la VoIP	12
Protocolos y Estándares VoIP	14
Calidad de Servicio.....	15
SNMP	21
Red de Telefonía Digital de Intercable	23
Regresión Lineal Múltiple	26
Sistema de Variables.....	33
Definición de Términos.....	33
III MARCO METODOLOGICO	37
Tipo de Investigación.....	37
Población	38
Muestra	38
Fases del Estudio.....	39
Fase 1. Diagnóstico.....	39
Fase 2. Revisión Bibliográfica.....	39
Fase 3. Construcción del Modelo	40
Fase 4. Evaluación del Modelo.....	40
IV ANALISIS DE LOS RESULTADOS	41
Construcción del Modelo	41
Análisis y Comparación de Resultados	53

V CONCLUSIONES Y RECOMENDACIONES	54
Conclusiones	54
Recomendaciones	56
VI SIMULACION DEL MODELO DE PREDICCIÓN DE CALIDAD DE SERVICIO PARA LA RED DE TELEFONIA DIGITAL DE INTERCABLE	57
REFERENCIAS BIBLIOGRAFICAS.....	63
ANEXOS	65

INDICE DE TABLAS

		pág.
TABLA		
1	La telefonía sobre IP	13
2	Operacionalización de las Variables.....	33
3	Data capturada el día lunes 16-07-07	43
4	Data capturada el día martes 17-07-07	43
5	Data capturada el día miércoles 18-07-07	43
6	Data capturada el día jueves 19-07-07.....	44
7	Data capturada el día viernes 20-07-07	44

INDICE DE FIGURAS

FIGURA		pág.
1	Metodología para la evaluación de servicios de Telecomunicación desde la perspectiva del usuario.....	18
2	Diseño y Desarrollo de un Sistema Experimental Cuantificador de Tráfico de Voz sobre IP (VoIP)	20
3	Identificadores de Objetos	22
4	Red Nacional Intercable.....	23
5	CS2000C.....	24
6	MS2010	24
7	MCS 5200	25
8	Proceso de una llamada	25
9	yi calculada Vs el error.....	52
10	Pantalla principal del software de simulación	57
11	Pantalla para cargar los datos de una llamada	58
12	Datos y Gráfica de QoS calculada y el error de una llamada	59
13	Datos y Gráfica de QoS para los datos del día lunes 16/07/07.....	59
14	Datos y Gráfica de QoS para los datos del día martes 17/07/07.....	60
15	Datos y Gráfica de QoS para los datos del día miercoles 18/07/07..	60
16	Datos y Gráfica de QoS para los datos del día jueves 19/07/07	61
17	Datos y Gráfica de QoS para los datos del día viernes 20/07/07	61

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIA Y TECNOLOGIA
MAESTRIA EN CIENCIAS DE LA COMPUTACION
MENCION REDES DE COMPUTADORAS**

**MODELO DE PREDICCIÓN DE CALIDAD DE SERVICIO PARA LA RED
DE TELEFONIA DIGITAL DE INTERCABLE**

Autor: Nayvic Piña

Tutor: Arsenio Pérez

RESUMEN

En este proyecto se elaboró un modelo para la predicción de QoS (Calidad de Servicio) en la red de telefonía digital de Intercable, a través del análisis de los parámetros que la afectan, como son el jitter, el delay y la pérdida de paquetes, definidos estos como estándares de la industria, con el fin de garantizar la comunicación y disponibilidad en la red. Los parámetros se recolectaron a través de lecturas del tráfico haciendo uso del Protocolo Simple de Administración de Red (SNMP) en el router principal de la red. El estudio se basó en una investigación de campo siguiendo la modalidad proyecto especial, se realizó un diagnóstico para determinar la necesidad de un modelo que permita predecir la QoS. Luego, a través de la revisión bibliográfica se estudiaron los parámetros establecidos por la industria para definir la QoS. Seguidamente se elaboró el modelo a través del método de regresión lineal múltiple ya que permite estudiar más de dos variables, y se definió la ecuación que simula la QoS. El estudio demostró que el modelo propuesto obtuvo una confiabilidad del 95%, se evaluó con los datos reales capturados, resultando la varianza o dispersión de los errores equivalente igual a 0.3019, valor que se encuentra muy cercano a cero. También se demostró que el modelo propuesto se adapta a la realidad estudiada ya que explica el 88.43% del total de la población. La importancia del estudio radica en los aportes del modelo sobre la capacidad de predicción de QoS, lo cual permite a las empresas detectar las fallas que se presenten y solucionarlas a tiempo con el fin de que los clientes reciban un buen servicio, garantizando la disponibilidad y control de los parámetros de QoS que exigen los organismos reguladores. El estudio deja abierto nuevas interrogantes de investigación sobre la QoS en VoIP.

Palabras claves: QoS, Voz sobre IP, Telefonía Digital.

INTRODUCCION

Actualmente se están desarrollando plataformas basadas en Voz sobre IP (VoIP) para transportar la voz en forma de paquetes sobre Internet y un aspecto importante a considerar en las Redes de Telefonía de VoIP es la Calidad de Servicio (QoS).

En este proyecto se propone desarrollar un modelo de predicción de QoS para la red de telefonía digital de Intercable tomando en cuenta que actualmente no dispone de ninguna herramienta que permita monitorear el estado de las llamadas ni los parámetros que afectan la QoS.

Se pretende desarrollar un modelo que beneficie a la empresa en cuanto a poder determinar de una manera más fácil la QoS que están recibiendo los clientes, que permita garantizar la disponibilidad del servicio, así como detectar fallas que se produzcan durante las llamadas, problemas como eco, retardos y ruidos que la afecten, así como establecer el control total en la administración.

Primero, se realizará un diagnóstico de la problemática que se presenta al no contar con una herramienta de monitoreo y no poder saber cual es la QoS brindada a los clientes; luego, en base a la investigación bibliográfica, se determinan los parámetros estándares de la industria que la afectan y finalmente se construyó el modelo y se evaluó atendiendo las necesidades presentadas y adaptando los costos de acuerdo a la disponibilidad.

Este estudio se ubica en una investigación de campo siguiendo la modalidad proyecto especial, y está estructurado como sigue:

El Capítulo I comprende: planteamiento del problema, justificación, los objetivos de la investigación, el alcance y las limitaciones que tendrá el desarrollo de la investigación.

El Capítulo II se refiere al Marco Teórico, el cual recopiló los antecedentes de la investigación así como toda la información bibliográfica y analítica que permitió

elaborar las bases teóricas que sustentan el estudio. Luego, se presenta la operacionalización de las variables que permite definir el rango de las mismas.

El Capítulo III correspondiente al Marco Metodológico especifica: tipo de investigación, diseño y fases del estudio.

El Capítulo IV describirá las bases estadísticas y matemáticas para la construcción del modelo, así como el análisis de los datos obtenidos.

El Capítulo V planteará un conjunto de Conclusiones y Recomendaciones sobre la investigación.

El Capítulo VI describe el software que simula el modelo desarrollado.

Finalmente se presenta la revisión bibliográfica y los anexos.

CAPITULO I

EL PROBLEMA

Planteamiento del Problema

Las empresas actualmente están desarrollando redes de tecnología de paquetes que permitan la convergencia de [voz](#) y [datos](#) aminorando las caídas de servicio, mejorando el control de tráfico y permitiendo el enlace con las redes telefónicas tradicionales. Así lo expresa Robins (2005) en un artículo publicado en la revista Internet Telephony, donde también hace referencia a los elementos que conforman las Redes IP para su funcionamiento como son: Los Teléfonos IP, Gateways, Gatekeeper, entre otros, junto con una serie de protocolos como el H.323 de la ITU-T, el protocolo de inicio de sesión (SIP), Megaco/H.248 y el protocolo en tiempo real (RTP) que cubren la mayor parte de las necesidades para la integración de voz.

También menciona que uno de los problemas más importantes que enfrentan este tipo de redes son los retardos de la transmisión de la voz por lo que hay que tomar en cuenta la Calidad de Servicio (QoS). La misma incluye la pérdida de paquetes, latencia, jitter y el eco lo que va a determinar el grado de satisfacción de un usuario con respecto al servicio que recibe. Con base a ello la calidad de servicio es la medida de la calidad de transmisión y de la disponibilidad de una red o grupo de redes. Un elevado nivel de calidad de servicio proporciona a las empresas considerables beneficios en cuanto a mercadeo, costos, motivación del cliente y diferenciación respecto a la competencia.

Intercable como empresa de telecomunicaciones está implementando telefonía digital. Este proyecto se inicio para permitir a los abonados realizar llamadas locales, larga distancia nacional e internacional y plantea entre sus retos brindar un servicio

con alta disponibilidad implementando mecanismos y técnicas que gestionen los recursos de la red como factores claves para la convergencia de voz y datos.

Por observación directa del investigador se pudo constatar que actualmente no se cuenta con una herramienta que permita monitorear los parámetros que inciden en la QoS, los valores que estos mantienen se observan a través de lecturas con el Protocolo Simple de Administración de Red (SNMP) de los eMTA (Adaptador Terminal Multimedia, Embedded Multimedia Terminal Adapter) o de forma más general en el router principal de la red.

Partiendo del contexto planteado se planteó construir un modelo que luego de monitorear los parámetros incidentes en la QoS, permita diagnosticar el estado de la red de telefonía y dar respuesta a las siguientes interrogantes:

¿Cuál será la situación actual de la red de telefonía digital de Intercable?

¿Cuáles son los parámetros que tienen mayor incidencia en la QoS?

¿En la red de telefonía digital de Intercable cual sería el modelo de predicción de QoS?

¿Cuál será el resultado de evaluar el modelo aplicado?

Objetivos de la Investigación

Para dar respuesta a las interrogantes se plantean los siguientes objetivos de investigación:

General

Desarrollar un modelo de predicción de Calidad de Servicio para la red de telefonía digital de Intercable.

Específicos

- Diagnosticar la situación de la red de telefonía digital de Intercable.

- Determinar los parámetros estándares que inciden en la QoS establecidos por la industria.
- Construir un modelo de predicción de QoS en función del análisis de los parámetros jitter, delay y pérdida de paquetes.
- Evaluar el modelo de predicción de QoS propuesto.

Justificación e Importancia

La QoS es un factor muy importante a medir en las redes actualmente, y es fundamental en el servicio de telefonía digital que va a prestar Intercable. Sería muy fácil ofrecer Calidad de Servicio si las redes nunca se congestionaran, pero como siempre existe este factor hay que implementar mecanismos que permitan brindar los servicios bajo cualquier circunstancia.

Intercable comenzó las comunicaciones de voz bajo el nombre de Intertel a mediados del año 2006 ofreciendo facilidades con clara orientación según el mercado objetivo. Los usuarios residenciales y corporativos cuentan con soluciones específicas a sus necesidades, por ejemplo: los usuarios residenciales pueden obtener un servicio a menor costo con sistemas integrados (videoconferencias, mensajería unificada, asistente personal virtual, entre otros) y los clientes corporativos pueden aumentar la productividad y movilidad de los empleados, mejorar los procesos de ventas, personalizar los sistemas integrados, entre otros.

Debido a esto se presenta la necesidad de monitorear la QoS ya que es lo que va a garantizar que el servicio se esté prestando en niveles óptimos, permitiendo detectar fallas fácilmente y el lugar donde se producen. También se pueden evitar problemas de eco y latencia, el operador de red puede obtener control de los recursos, ofrecer garantía de servicio y diferenciación de tráfico.

La QoS es un factor clave para que este producto alcance éxito, los suscriptores van a mantener sus contratos de acuerdo al servicio que se les brinde. Partiendo de

esta idea es necesario monitorear una serie de parámetros, garantizar la disponibilidad de la red y la calidad percibida por el cliente.

En este tipo de soluciones la QoS constituye un factor muy importante ya que va a determinar la satisfacción del usuario al poder realizar llamadas libres de eco, retardos y cualquier tipo de perturbaciones que interfieran en las llamadas.

Con base a los planteamientos expuestos se justifica plenamente el desarrollo de un modelo de predicción de QoS ya que en Intercable no se cuenta con una herramienta que permita monitorear los parámetros que la involucran y así poder determinar cuando van a existir fallas en las comunicaciones telefónicas. En general, es uno de los temas mas importantes en esta nueva etapa ya que es el factor determinante para brindar un mejor servicio a los clientes y competir con otras empresas de telecomunicaciones que también están desarrollando su plataforma tecnológica.

Este estudio podrá servir de base a otras investigaciones relacionadas con la evaluación de QoS como herramienta de control en la red.

Alcances y Limitaciones

Alcance

Esta investigación tuvo como primer alcance monitorear los parámetros dependientes de la QoS, jitter, delay y pérdida de paquetes en la red de telefonía digital de Intercable.

Luego mediante el método de regresión lineal múltiple se pudo obtener un modelo que relacionando las variables determina la calidad de las llamadas.

Esto permitió simular a través de un software construido en Visual Basic 6.0 introducir los datos capturados para estimar la QoS de las llamadas que se procesan en esta red de telefonía.

Limitaciones

Para monitorear los datos, actualmente en Intercable se utiliza SNMP el cual toma la información de una forma engorrosa de entender, por lo que luego de realizar estas lecturas se construyeron tablas con los datos un poco más lógicas.

Estas lecturas fueron tomadas por un administrador de red que labora en dicha empresa ya que no se permitía el acceso al investigador a la sala de comunicaciones principal.

CAPITULO II

MARCO TEORICO

Antecedentes de la Investigación

Anteriormente las llamadas, especialmente las de larga distancia, eran muy costosas por lo que muy pocas personas podían usar el servicio. Así lo afirman en la revista Telefonía Web (2007) en el artículo “*Reduce la factura del Teléfono llamando por Internet*” donde explican que hoy en día las comunicaciones son gratis gracias a la tecnología de Voz sobre IP.

Acquatella y Montenegro (2002), en su proyecto de investigación “*Diseño para la implementación de Voz sobre IP en la banda MMDS*”, definen los equipos para implementar VoIP y miden su capacidad de transmisión para la red de acceso en la parte inalámbrica (MMDS, Multichannel Multipoint Distribution Service), para así poder prestar el servicio de llamadas tomando en cuenta los parámetros capacidad, velocidad de CPU, adaptabilidad, escalabilidad, fiabilidad, modularidad y calidad de servicio (QoS). Este estudio se basó en el funcionamiento de VoIP para poder dimensionar y medir el desempeño de los equipos escogidos y así poder transmitir voz a través de una red IP. El trabajo desarrollado se relaciona con este proyecto en la necesidad de conocer los componentes que intervienen en las redes IP y estudiar las variables que afectan la calidad de servicio.

Barrios (2004), en su tesis de grado “*Estudio y definición de los parámetros e indicadores de calidad de servicio de Internet a ser aplicados en Venezuela*”, define los parámetros e indicadores para medir la calidad de servicio de Internet así como la metodología de medición que deberían emplear los operadores. Para esto analiza la

latencia, número de intentos para la conexión, tasa de pérdida de paquetes, tiempo promedio de establecimiento de la conexión, entre otros, para lo cual parte de la revisión de normas y experiencias internacionales relativas a la QoS en Internet, factor clave para las comunicaciones de VoIP y que van a determinar la calidad de la llamada que transcurre por este medio, lo que demuestra la relación con esta investigación.

Bellido (2004), en su tesis doctoral *“Contribución a las metodologías para la evaluación de la calidad de servicio en redes heterogéneas”*, realizó un estudio de Calidad de Servicio sobre las Redes IP y contribuye con una metodología de evaluación para comparar el nivel de servicio de los distintos proveedores tomando la información de usuarios finales, es decir, su objetivo contribuye a comprender mejor las necesidades de los clientes y evaluar su grado de satisfacción. Se basa en los componentes del modelo de QoS, como son calidad de servicio percibida (éxito en la conexión, calidad de audio y video, disponibilidad/fiabilidad del servicio), prestaciones funcionales de la red (pérdida de paquetes, retardo de extremo a extremo) y Service Level Agreement (SLA). En general, la metodología incluye la definición de las clases de servicio y la propuesta de un sistema de evaluación y seguimiento de QoS donde se evalúa la calidad desde el proveedor (teniendo en cuenta la relación entre servicio ofrecido, servicio prestado y calidad percibida), la calidad desde el usuario y la calidad desde el sistema de evaluación. Este proyecto tiene estrecha relación con esta investigación al analizar la funcionalidad de la red ya que una vez más se estudian los parámetros estándares de la industria que determinan la QoS.

López y otros (2005), en su investigación *“Metodología para la evaluación de servicios de Telecomunicación desde la perspectiva del usuario”*, propone un sistema para la evaluación de la calidad que prestan los distintos proveedores desde la perspectiva de los usuarios basándose en la calidad percibida por los clientes y las funciones que deben desempeñar los agentes que prestan el servicio. En primera instancia, realizan la clasificación de los servicios de telecomunicaciones, luego,

clasifican los factores que afectan la QoS, entre los que estudian la funcionalidad de la red y determinan que la calidad de servicio percibida es la adecuada, a través de la medición de los parámetros: pérdida de paquetes, retardos, etc. Esto se relaciona con el proyecto en desarrollo al estudiar el comportamiento de los mismos y su relación con la QoS prestada.

Belzarena y otros (2006), en su proyecto “*Metronet: software para medición de calidad de servicio en voz y video*”, desarrollan un software que tiene como objetivo estimar la QoS para aplicaciones multimedias en Internet. En la primera etapa desarrollan un modelo tomando en cuenta que los paquetes pueden tomar diferentes rutas en una red e incluso pasar por activos de otros proveedores de servicio, por lo que desarrollan un modelo para el tráfico multimedia y uno para el cross traffic, tomando en cuenta la capacidad y tamaño del buffer de los enlaces a lo largo de un camino. Usando la estimación del cross traffic miden los parámetros de QoS (retardo, pérdidas, jitter, etc.) y los relacionan a través de una función para obtener la QoS para una aplicación multimedia. Posteriormente desarrolla el software Metronet el cual es un modelo cliente servidor, quien quiere medir la QoS actúa como cliente y el proveedor de servicio como servidor. A diferencia, este trabajo de investigación monitorea los parámetros de QoS de voz en el activo principal de la red, y los relaciona a través de un modelo el cual determina la QoS de una llamada y permite a los especialistas monitorear los valores de los mismos. El proyecto “*Metronet*” tiene relación con la presente investigación ya que se estudiaron y construyeron modelos estadísticos relacionados con el tráfico de la red, en función de los parámetros dependientes de la QoS y a la vez se creó el software basado en dichos modelos, mas no permite determinar la calidad de cada llamada como se simula en este trabajo.

Se puede observar que las investigaciones antes mencionadas guardan relación directa con el desarrollo de este proyecto de investigación al estudiar los parámetros que garantizan QoS como parte funcional de los sistemas de comunicación por Internet.

Bases Teóricas

Para el desarrollo de este proyecto de investigación se va a realizar un análisis de la teoría relacionada con VoIP y análisis estadísticos con regresión lineal múltiple, basada en los siguientes temas fundamentales: Inicios de la VoIP, componentes de la VoIP, protocolos y estándares VoIP, calidad de servicio, SNMP (Protocolo Simple de Gestión de Red), Red de Telefonía Digital de Intercable y otros.

Inicios de la VoIP

La VoIP se define como una tecnología que permite la transmisión de la voz a través de redes IP en forma de paquetes de datos. Así lo explican en la página Web Recursos VoIP (2006) donde afirman que la Telefonía IP es una aplicación inmediata de esta tecnología, de forma que permita la realización de llamadas telefónicas ordinarias sobre redes IP u otras redes de paquetes utilizando un PC, gateways y teléfonos estándares.

Ares (2006), en su publicación "*Telefonía-IP*" expresa que la VoIP en sus inicios se implementó para reducir el ancho de banda mediante compresión vocal, aprovechando los procesos de compresión diseñados para sistemas celulares en la década de los años 80 logrando reducir los costos en el transporte internacional.

Rojano(2006) en la ponencia "*Introducción a la VoIP con Linux Asterisk*", expresa que la VoIP tiene sus orígenes desde 1.995 donde afirma que comenzó con pequeñas aplicaciones gratuitas y de código abierto a raíz de la posibilidad de enviar pequeños fragmentos de voz codificados.

En 1.996, aparecen los protocolos de comunicaciones con aplicaciones como ICQ y Netmeeting y en 1.997 surge el protocolo H.323 para voz y video aunque con poca calidad por limitaciones de ancho de banda, pero impulsa a desarrollar hardware y software que actúan como centrales de VoIP.

Entre 1.998 y 1.999 surge la banda ancha y la VoIP se piensa como una herramienta a futuro para telefonía a muy bajo costo, así Netmeeting permite conexiones con un servidor H.323 y la aplicación de voz y video más conocida para el momento es CU-SeeMe. Comienza SIP como una evolución de H.323 y Mark Spencer da a conocer Asterisk el cual para el 2.000 empieza como software abierto a través de un grupo de seguidores.

A partir del 2.001 hasta la actualidad se convierte en una importante herramienta para VoIP, convirtiendo a la empresa Linux-support en Digium como especialista en venta de hardware especial para Asterisk.

Es importante como se realiza la comunicación a través de una red IP; en la misma referencia Web Recursos VoIP (2006) explica que cuando hacemos una llamada telefónica la voz se digitaliza, se comprime y se envía en paquetes de datos IP. Estos paquetes se envían a través de Internet a la persona con la que estamos hablando. Cuando alcanzan su destino, son ensamblados de nuevo, descomprimidos y convertidos en la señal de voz original.

Componentes de la VoIP

Las redes de VoIP según Huidobro (2006) en su publicación web “*La telefonía sobre IP*” contienen cuatro componentes:

- 1. Cliente Multimedia:** Se trata normalmente de un PC multimedia.
- 2. Gateways de Voz/IP:** Es el componente clave de una solución de voz sobre IP al facilitar la conversión de las llamadas telefónicas convencionales al mundo de IP.
- 3. Gateway H.323/H.320:** Básicamente, realiza la conversión entre estos dos formatos de forma que los terminales H.323 se pueden comunicar con equipos RDSI de videoconferencia, que pueden formar parte de la red corporativa o estar situados en la red pública.

4. Gateway H.323/RTB (Voz sobre IP): Posibilitan las comunicaciones de voz entre los terminales H.323 y los teléfonos convencionales, estén en la red corporativa o en la red pública.

5. Gatekeeper: El Gatekeeper es un punto central de control en una red H.323, proporcionando servicios de control de llamada, traducción de direcciones y control de admisión. Además facilita el control del ancho de banda utilizado y localiza los distintos gateways y MCU's cuando se necesita.

6. Gatekeeper H.323: Está siempre presente para controlar las llamadas en la Intranet Pública (o red corporativa). Todos los elementos de red de MMTS (terminales, Gateways, MCU) tienen que usar el Gatekeeper como punto intermedio para la señalización. De esta forma se tiene un control de los accesos, seguridad, movilidad del usuario, y tarificación si se da el caso.

7. MCU para H.323 y T.120: Se utiliza cuando han de intervenir más de dos partes en una conferencia. La MCU (Multimedia Conference Unit) es responsable de controlar las sesiones y de efectuar el mezclado de los flujos de audio, datos y video.

En la tabla siguiente se muestran los componentes ubicados en el modelo OSI (Open System Interconnection):

Tabla 1
La telefonía sobre IP

CAPAS	COMPONENTES
PRESENTACION	G.729(A)/G.723.1/G.711
SESION	H.323/SIP/SDP
TRANSPORTE	RTP/UDP/RSVP
RED	IP/WFQ/IP-prec
ENLACE	C RTP
FISICA	

Fuente: <http://www.recursosvoip.com/colabora/teleip1.php>

Protocolos y Estándares VoIP

En el documento en línea “*Descripción técnica detallada sobre Voz sobre IP (VOIP)*” se describen los estándares y protocolos de la VoIP según sus funciones:

1. Direccionamiento:

a. **RAS (Registration, Admission and Status):** Protocolo de comunicaciones que permite a una estación localizar otra a través de el Gatekeeper.

b. **DNS (Domain Name Service):** Servicio de resolución de nombres en direcciones IP con el mismo fin que el protocolo RAS pero a través de un servidor DNS.

2. Señalización:

a. **Q.931:** Señalización inicial de llamada.

b. **H.225:** Control de llamada: señalización, registro y admisión, y paquetización / sincronización del stream (flujo) de voz.

c. **H.245:** Protocolo de control para especificar mensajes de apertura y cierre de canales para streams de voz.

d. **H.248:** También conocido como Megaco o MGCP (protocolo de control gateway media), define el mecanismo necesario de llamada para permitir a un controlador Media Gateway el control de puertas de enlace para soporte de llamadas de voz/fax entre redes.

e. **H.323:** Es el estándar creado por la Unión Internacional de Telecomunicaciones (ITU) que se compone por un protocolo sumamente complejo y extenso, el cual además de incluir la voz sobre IP, ofrece especificaciones para videoconferencias y aplicaciones en tiempo real, entre otras variantes.

f. **Protocolo de Inicio de Sesión (SIP):** Fue desarrollado por la IETF (Internet Engineering Task Force) específicamente para telefonía IP, que a su vez toma ventaja de otros protocolos existentes para manejar parte del proceso de conversión, situación que no se aplica en H.323 ya que define sus propios protocolos bases.

g. Compresión de Voz:

h. Requeridos: G.711 y G.723

i. Opcionales: G.728, G.729 y G.722

3. Transmisión de Voz:

a. UDP: La transmisión se realiza sobre paquetes UDP, aunque no ofrece integridad en los datos, el aprovechamiento del ancho de banda es mayor que con TCP.

b. RTP (Real Time Protocol): Maneja los aspectos relativos a la temporización, marcando los paquetes UDP con la información necesaria para la correcta entrega de los mismos en recepción.

4. Control de la Transmisión: Se usa el protocolo de control en tiempo real (RTCP), para detectar situaciones de congestión de la red y tomar, en su caso, acciones correctoras.

Calidad de Servicio (QoS)

Al implementar Voz sobre IP se presenta la necesidad de medir los parámetros de QoS necesarios para garantizar la comunicación eficiente entre usuarios, de aquí la necesidad de estudiar un modelo que permita establecer en que niveles es óptimo y en cuales se pueden presentar fallas. Según Davidson (2001), para esto es necesario hacer un estudio de los parámetros que afectan la QoS y como influyen en su funcionamiento, así como otros conceptos de VoIP relacionados.

La QoS se refiere a la Clase de Servicio (CoS) y al Tipo de Servicio (ToS) para conseguir el ancho de banda y la latencia para una aplicación particular. La CoS permite agrupar los flujos de paquetes y el ToS es un campo en la cabecera IP para definir la clase de servicio. Para implementar QoS se deben cumplir un grupo de requisitos de servicio al transportar un flujo de datos controlando parámetros como jitter, delay, pérdida de paquetes, regulación de tráfico entre otros.

Según Bravo (2005) la QoS es la medida de la calidad de transmisión y de la disponibilidad de una red o grupo de redes. También la define como un grupo de requisitos de servicio que deben cumplirse por la red al transportar un flujo de datos. Las funciones de QoS IP están diseñadas para proporcionar servicios diferenciados y/o garantizados proporcionando al operador mecanismo de control de los recursos.

Al implantar QoS se intenta que ciertos parámetros de desempeño se encuentren dentro de valores para que la calidad percibida sea la correcta. Los parámetros para garantizar el desempeño de la conexión los define de la siguiente manera:

1. Ancho de Banda: Es utilizado para describir la tasa de bits por unidad de tiempo de un medio determinado, de un protocolo o una conexión. Describe el “tamaño del canal” requerido por la aplicación para comunicarse sobre la red.

2. Retardo y Jitter: Bastero (2005) define el retardo o latencia como:

“Es el tiempo que necesita la voz para viajar desde el micrófono de un teléfono al auricular del teléfono remoto, es la suma del retardo que introduce el CODEC seleccionado, el buffer del jitter en el teléfono y el trayecto utilizado para transportar los paquetes a través de la red”.

El retardo está compuesto de:

Retardo de Transmisión: Es el tiempo que tarda un dispositivo en enviar el paquete completo a través de la interfaz de salida. Depende del ancho de banda del enlace así como el tamaño del paquete.

Retardo de Propagación: Es el tiempo que toma a un bit de ir desde el transmisor hasta el enlace del receptor. Para enlaces WAN, los retardos de propagación en el orden de milisegundos son normales.

Retardo de Conmutación: Es el tiempo que toma para un dispositivo el transmitir un paquete después de recibirlo de otra fuente. Es un valor bajo en el orden de los micro segundos.

Todos los paquetes de un flujo no experimentan el mismo retardo en la red. Si la red no está congestionada, el retardo de transmisión y propagación serán los factores mas influyentes. Si la red está congestionada comenzará a producirse una variación

del retardo entre diferentes paquetes. La variación del retardo de los paquetes se denomina Jitter y el mismo autor afirma:

“El Jitter de los paquetes es importante porque éste estima el máximo retardo posible en la recepción de los paquetes respecto al retardo individual. Un receptor dependiendo de la aplicación puede sobreponerse al Jitter añadiendo un buffer de recepción que pueda almacenar los paquetes hasta el valor extremo del Jitter”. (p. 43)

3. Pérdida de Paquetes: Especifica el número de paquetes que se pierden en la red durante la transmisión. El descarte de paquetes en los puntos de congestión de la red y los paquetes que se corrompen en el medio causan la pérdida de paquetes. El descarte de paquetes ocurre generalmente en los puntos donde hay congestión en los que los paquetes que llegan exceden el límite máximo del tamaño de la cola de salida, también puede ocurrir porque los buffers de entrada son insuficientes.

También define los siguientes requisitos básicos de la VoIP:

1. Las pérdidas no deben superar el 1%.
2. El retardo de una vía no debe superar los 150 ms.
3. El jitter debe estar por debajo de 30 ms.
4. El ancho de banda varía entre 21 y 320 kbps por llamada, dependiendo del codec, de la tasa de muestreo y del protocolo de capa 2.

Entre los beneficios de la QoS IP menciona:

1. Permite a la red soportar los requisitos de las aplicaciones multimedia.
2. Proporciona al operador de red el control de los recursos de red y su uso.
3. Proporciona garantías de servicio y diferenciación de tráfico a través de la red. Esto se requiere para llevar tráfico de voz, datos y video sobre la misma red.
4. Permite a los proveedores de servicio ofrecer servicios Premium en conjunto con la clase de servicio Best Effort. Por ejemplo se podrían ofrecer servicios Oro, Plata y Bronce con distinto soporte QoS.
5. Permite que la red conozca las aplicaciones que viajan a través de ella, en las cuales los servicios de red manejan los paquetes basados en la información de aplicación que aparece en la cabecera de los paquetes.

6. Juega un rol esencial en servicios tales como las redes privadas virtuales (VPN).

En cuanto a los niveles de QoS IP según la capacidad de las redes de entregar el servicio requerido por una cierta aplicación con algún grado de control sobre las medidas de rendimiento (ancho de banda, retardo/jitter y pérdidas) menciona las siguientes:

1. Servicio Best Effort (Mejor Esfuerzo): Se ofrece conectividad básica sin garantías.

2. Servicio Diferenciado: En este tipo de servicios el trabajo se agrupa en clases dependiendo de sus requisitos de servicio. Cada clase de tráfico es tratada de forma distinta por la red y es servida de acuerdo a los mecanismos de QoS configurados para la clase. Los servicios diferenciados no ofrecen garantías de servicio, solo ofrece diferenciación de una clase frente a otra.

3. Servicio Garantizado: Un servicio que requiere la reserva de recursos para asegurar que la red cumple con los requisitos específicos de un flujo de tráfico.

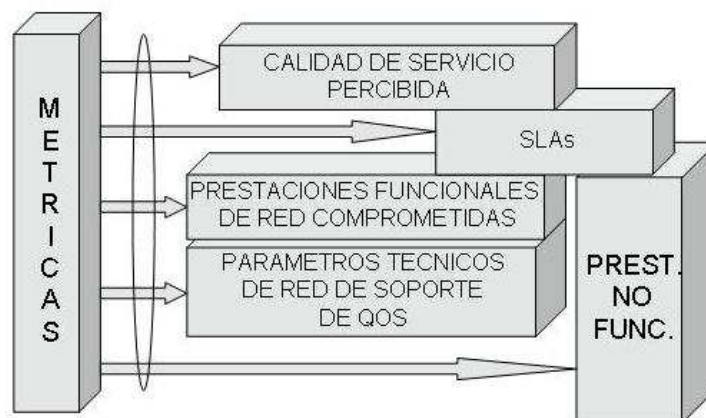


Figura 1. Metodología para la evaluación de servicios de Telecomunicación desde la perspectiva del usuario. Fuente: <http://jungla.dit.upm.es/~jlopez/publicaciones/telecomid04lbt.pdf> (p.2)

Según el estudio de López y otros (2005) la clasificación de QoS involucra los niveles que se muestran a continuación:

1. Calidad de Servicio Percibida: Parámetros de cada servicio que el usuario percibe y puede comprobar y determina su mayor o menor satisfacción, por ejemplo

éxito en la conexión, calidad del audio y video, disponibilidad y fiabilidad del servicio.

2. Prestaciones Funcionales de Red: Especificaciones técnicas de la red cuyo cumplimiento implica que la QoS percibida es la adecuada, por ejemplo pérdida de paquetes y retardos.

3. Service Level Agreement (SLA): Contrato entre el cliente y el proveedor, métodos de cumplimiento de los mismos y penalizaciones al incurrir en incumplimiento.

4. Parámetros Técnicos de Red de Soporte: Parámetros internos de la red que determinan el comportamiento de las prestaciones funcionales y determinan la calidad del transporte, por ejemplo probabilidad de congestión, conmutación y enrutamiento.

5. Prestaciones no funcionales: Los parámetros que definen provisión, gestión y mantenimiento que reflejan el grado de satisfacción del cliente con respecto al proveedor, por ejemplo tiempo de demora en instalar el servicio o de atender quejas.

6. Métricas: Posee un nivel general donde se definen parámetros, métodos de medida y muestreo y otro particular para cada uno de los niveles anteriores donde se especifica calidad, definición de valores, umbrales y procedimientos de medida.

La calidad de la voz se puede medir de manera subjetiva, como el ser humano la percibe y objetiva a través de instrumentos de medición. Una manera objetiva es a través del Mean Opinión Store (MOS) con una puntuación de 1 a 5 y una manera objetiva se realiza mediante la recomendación P.861 de ITU-T mediante la percepción Perceptual Speech Quality Measurement (PSQM).

Otro factor importante que afecta la QoS es el eco, cuando escuchamos nuestra voz en el auricular con retardo produce interrupciones en la conversación. Estos problemas generalmente se resuelven con canceladores de eco. Un proveedor garantiza QoS cuando se garantiza el valor de los parámetros que ofrece la red, si no decimos que lo que ofrece es un servicio best effort.

El contrato que especifica los parámetros de QoS acordados entre el proveedor y el usuario (cliente) se denomina SLA (Service Level Agreement), y especifica los valores típicos que el proveedor debe cumplir para garantizar QoS. Por ejemplo podría acordar la disponibilidad un 99%, pérdida de paquetes 0.1%, delay 82 ms y jitter 20 ms.

Al respecto Pons (2002), diseñó el sistema cuantificador de tráfico de voz (PC-Voz) con una estructura modular de tal manera que es posible cuantificar otros protocolos de VoIP sustituyendo el Módulo de Análisis de Paquetes donde se identifican los factores del protocolo respetando las entradas y salidas de los otros módulos.

En el Gráfico 2 se muestra el diseño estándar de PC-VOZ, está compuesto por dos aplicaciones corriendo en paralelo, una es el Kernel del Programa Cuantificador de Voz (KPC-VOZ) que se encarga de capturar los paquetes y filtrarlos, los cuales son una cadena de caracteres que especifican parámetros que son evaluados por la librería de captura y que restringen el tráfico. En un archivo de texto (log) se van almacenar todos los paquetes capturados y en la base de datos de una manera ordenada las llamadas origen y destino, así como los usuarios. La otra aplicación es el módulo de seguimiento de actividad (MoSAc) el cual es opcional.

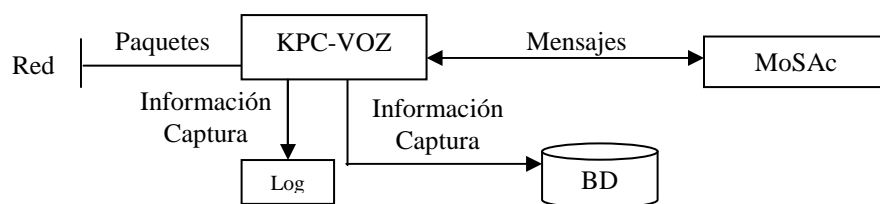


Figura 2. Diseño y Desarrollo de un Sistema Experimental Cuantificador de Tráfico de Voz sobre IP (VoIP). Fuente: Pons Daniel (p.35)

SNMP (Protocolo Simple de Gestión de Red)

SNMP es un protocolo de capa de aplicación que fue diseñado para facilitar el intercambio de información entre los dispositivos de una red, así lo definen en la Guía del segundo año CCNA 3 y 4. También expresan que usando SNMP y gracias al

número de paquetes de datos por segundos enviados a una interfaz o conexión TCP abierta, los administradores de la red pueden controlar fácilmente su rendimiento y resolver los problemas que puedan aparecer.

El modelo organizativo para la administración de las redes basadas en SNMP está compuesto por cuatro elementos: Administración de estación, Administración de agente, MIB (Base de información de administración, Management Information Base) y Protocolo de administración de red.

El NMS (Sistema de administración de la red, Network Management System), es una estación de trabajo típica que ejecuta un sistema operativo corriente, que dispone de una cantidad de RAM suficiente como para almacenar todas las aplicaciones de administración que se están ejecutando al mismo tiempo.

El NMS lleva a cabo la monitorización recuperando los valores desde la MIB y puede hacer que se lleve a cabo una acción en un agente o cambiar la configuración de otro. El protocolo SNMP opera sobre el UDP Protocolo de Datagrama de Usuario, User Datagram Protocol), usando los puertos 151/152 y esta basado en el intercambio de los siguientes tres tipos de mensajes:

- 1. Get:** Habilita a la estación de administración a recuperar el valor de los objetos MIB desde el agente.
- 2. Set:** Habilita a la estación de administración a establecer el valor de los objetos MIB en el agente.
- 3. Trap:** Habilita al agente a notificar los eventos significativos a la estación de administración.

El NMS se encuentra habitualmente en una estación de trabajo, pero puede implementarse en distintos sistemas e incluye una colección de software, llamado aplicación de administración de red. Responde a los comandos de usuario y a los emitidos para dirigir a los agentes de administración a lo largo de la red, lo cuales, son las plataformas y dispositivos claves de la misma (otros host, routers, puentes y hubs que están equipados con SNMP para que puedan ser administrados). Este último

responde a las peticiones de información y demandan acciones desde el NMS y los datos de cada uno se almacenan en su MIB particular.

El agente es una función software que está integrada en la mayor parte de los dispositivos de red como routers, switches, hubs, impresoras y servidores. El agente es responsable del procesamiento de las peticiones SNMP del administrador. También se encarga de la ejecución de las rutinas de mantenimiento de las variables definidas en las distintas MIB soportadas.

La interacción entre el administrador y el agente se realiza a través del protocolo de pregunta y respuesta o SNMP. Lo antes expuesto se ilustra en el Gráfico 3.

Un agente puede controlar: Número y estado de sus circuitos virtuales, número de ciertos tipos de mensajes de error recibidos, número de bytes y paquetes entrantes y salientes del dispositivo, longitud máxima de la cola de salida (para routers y otros dispositivos de internetworking), mensajes de difusión enviados y recibidos, interfaces de red que han caído y las que se han activado.

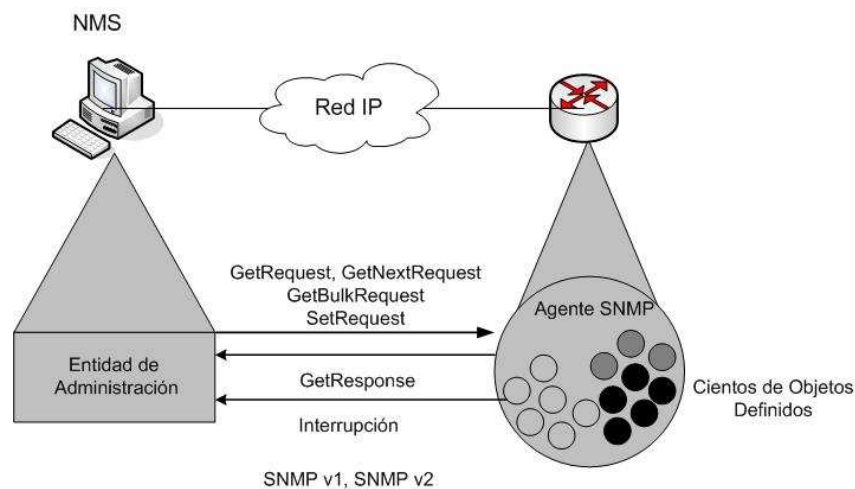


Figura 3. Identificadores de Objetos. Fuente: Guía del segundo año CCNA (p. 566)

Red de Telefonía Digital de Intercable

La solución de red escogida fue la propuesta por la empresa Nortel Network la cual incluye una serie de componentes de telefonía que permiten la comunicación.

En el *Gráfico 4* se muestra el diagrama de la Red Nacional de Intercable:

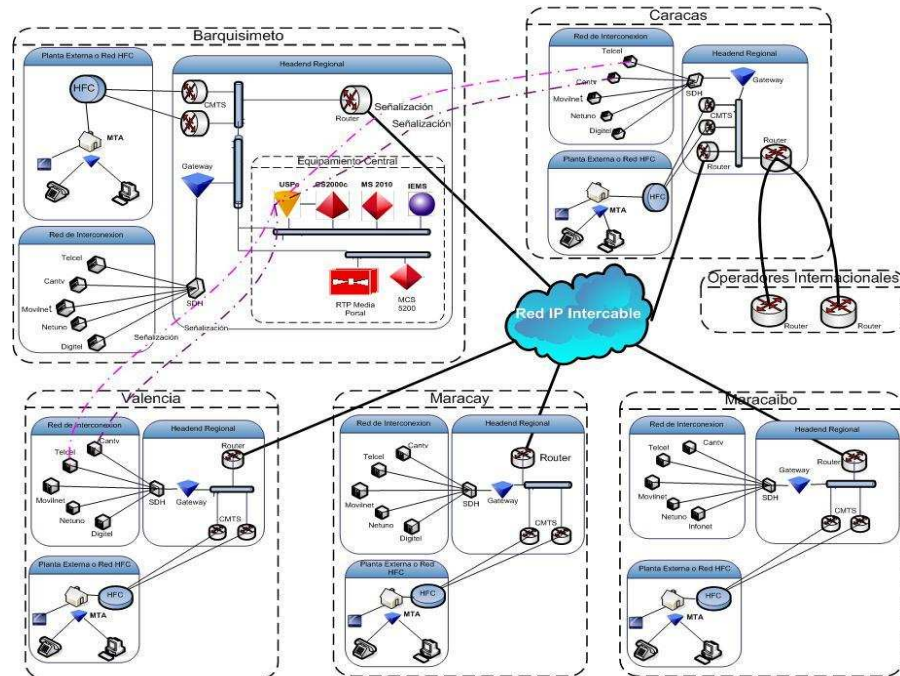


Figura 4. Red Nacional Intercable. Fuente: Red Digital de Intercable (p.22)

El núcleo principal se encuentra en Barquisimeto donde están conectados los equipos siguientes dispuestos para telefonía:

CS2000C (Communication Server 2000 - Compact): Softswitch Superclass de Nortel Network, que proporciona control de las llamadas, traducción, enrutamiento, administración de conexión, facturación y señalización centralizada. Opera con los protocolos H.323, H.248, Media Gateway Control Protocol (MGCP), Network-based Call Signaling (NCS) y Protocolo de Inicio de Sesión (SIP).

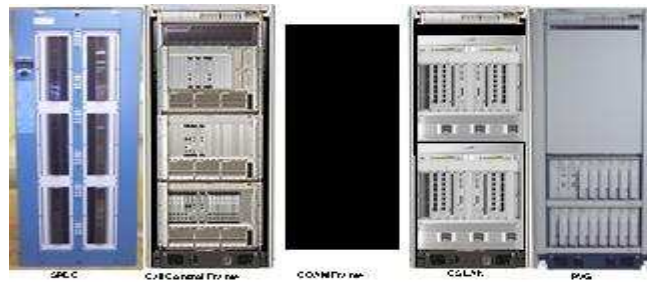


Figura 5. CS2000C. Fuente:

<http://support.nortel.com/go/main.jsp?cscat=CATEGORY&catOID=-9602>

IEMS (Integrated Element Management System): Sistema operacional integrado que proporciona un marco de las operaciones de la red para simplificar la administración de elementos. La administración centralizada simplifica la autenticación, autorización, administración y el registro.

MS2010 (Media Server 2000): Ofrece capacidad de anuncio en una red de paquetes y entrega de servicios avanzados de audio. Graba 20 minutos en tiempo real y 40 minutos sin tiempo real. Protocolo de Interfaz H.248.



Figura 6. MS2010. Fuente:

<http://support.nortel.com/go/main.jsp?cscat=CATEGORY&catOID=-9602>

MCS 5200 (Multimedia Communications Server 5200): Proporciona aplicaciones para empresas basadas en SIP, permite a los usuarios control de las comunicaciones, redirecciona las llamadas entrantes basado en el identificador de llamadas, hora y fecha a cualquier dispositivo conveniente, habilita con un click sobre el contacto la llamada, facilita el intercambio de información, imágenes y texto entre individuos geográficamente dispersos, permite llamadas con video desde el escritorio del usuario sin necesidad de procedimientos complicados, permite el reenvío, transferencia de

llamadas y llamadas en espera, sobre la infraestructura VoIP. Posee servicio de MSN, telefonía, Webcolaboration y correo de voz.



Figura 7. MCS 5200. Fuente:

<http://support.nortel.com/go/main.jsp?cscat=CATEGORY&catOID=-9602>

USP (Punto de Señalización Universal): Gateway de señalización para la interconexión entre redes TDM y los elementos del CS2000C.

Al integrar todos estos componentes en la red podemos observar en la *Figura 5* como se efectúa la llamada, la misma pasa a través de los servidores en el equipamiento central principal ubicado en la Sede principal de Barquisimeto y llega al CMTS BSR 64000 que es el router encargado del manejo de los CABLE MODEM para intercambiar señales digitales sobre la red HFC, administra el espectro para garantizar calidad de servicio y entrega confiable, luego está conectado el teléfono en el MTA (Multimedia Terminal Adapter) que es una interfaz de red, codecs, señalización y encapsulación de funciones para el transporte de VoIP.

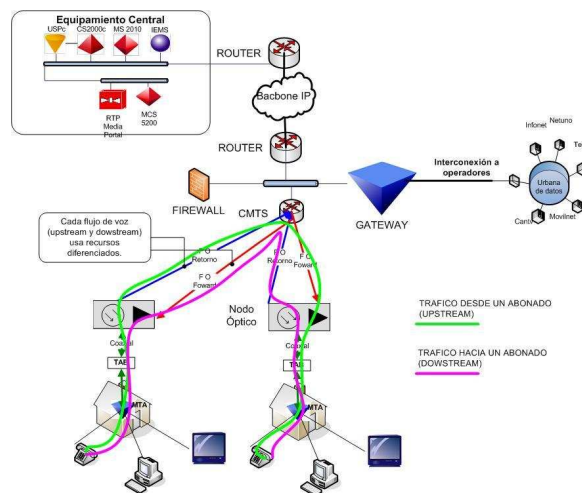


Figura 8. Proceso de una Llamada. Fuente: Red Digital de Intercable (p.35)

Regresión Lineal Múltiple

En la mayor parte de los problemas de investigación en que se aplica el análisis de regresión, se requiere más de una variable independiente en el modelo de regresión. Así lo afirma Meier(2004), y expresa también que cuando los coeficientes son lineales recibe el nombre de modelo de regresión lineal múltiple. Para el caso de k variables independientes x_1, x_2, \dots, x_k , la media de $Y| x_1, x_2, \dots, x_k$ está dada por el modelo de regresión lineal múltiple:

$$\mu_{Y| x_1, x_2, \dots, x_k} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

La respuesta estimada se obtiene de la ecuación de regresión muestral:

$$\hat{y} = b_0 + b_1 x_1 + \dots + b_k x_k$$

donde cada coeficiente de regresión B_i se estima mediante b_i de los datos muestrales con el método de los mínimos cuadrados. Igual que en el caso de una sola variable independiente con frecuencia, el modelo de regresión lineal múltiple puede ser una representación adecuada de una estructura más complicada dentro de ciertos rangos de las variables independientes.

1. Estimación de los coeficientes: Según el mismo autor se deben obtener los estimadores de mínimos cuadrados de los parámetros $\beta_0, \beta_1, \dots, \beta_k$ ajustando el modelo de regresión lineal múltiple:

$$\mu_{Y| x_1, x_2, \dots, x_k} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

a los puntos de datos

$$\{(x_{1i}, x_{2i}, \dots, x_{ki}, y_i)\}; \quad i=1, 2, \dots, n \quad \text{y} \quad n > k$$

Donde y_i es la respuesta observada para los valores $x_{1i}, x_{2i}, \dots, x_{ki}$ de las k variables independientes x_1, x_2, \dots, x_k . Cada observación $(x_{1i}, x_{2i}, \dots, x_{ki}, y_i)$ satisface la ecuación:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \varepsilon_i$$

o

$$y_i = b_0 + b_1 x_{1i} + b_2 x_{2i} + \dots + b_k x_{ki} + e_i$$

donde ε_i y e_i son los errores aleatorios y residual, respectivamente, asociados con la respuesta y_i . Al utilizar el concepto de mínimos cuadrados para llegar a los estimadores de $b_0 + b_1X_{1i} + b_2X_{2i} \dots + b_kX_{ki} + e_i$ se minimiza la expresión:

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - b_0 - b_1X_{1i} - b_2X_{2i} \dots - b_kX_{ki})^2$$

Diferenciando SSE cada vez con respecto a $b_0, b_1, b_2, \dots, b_k$, e igualando a cero, se genera el conjunto de $k+1$ ecuaciones normales:

$$\begin{aligned} nb_0 + b_1 \sum_{i=1}^n X_{1i} + b_2 \sum_{i=1}^n X_{2i} + \dots + b_k \sum_{i=1}^n X_{ki} &= \sum_{i=1}^n y_i \\ b_0 \sum_{i=1}^n X_{1i} + b_1 \sum_{i=1}^n X_{1i}^2 + b_2 \sum_{i=1}^n X_{1i} X_{2i} + \dots + b_k \sum_{i=1}^n X_{1i} X_{ki} &= \sum_{i=1}^n X_{1i} y_i \\ \cdot & \cdot \cdot \cdot \cdot \\ \cdot & \cdot \cdot \cdot \cdot \cdot \\ \cdot & \cdot \cdot \cdot \cdot \cdot \\ b_0 \sum_{i=1}^n X_{ki} + b_1 \sum_{i=1}^n X_{ki} X_{1i} + b_2 \sum_{i=1}^n X_{ki} X_{2i} + \dots + b_k \sum_{i=1}^n X_{ki}^2 &= \sum_{i=1}^n X_{ki} y_i \end{aligned}$$

Estas ecuaciones se pueden resolver para $b_0, b_1, b_2, \dots, b_k$, por cualquier método apropiado para resolver sistema de ecuaciones lineales.

2. El modelo de regresión lineal mediante matrices: Al ajustar un modelo de regresión lineal múltiple, en particular cuando el número de variables excede de dos, la teoría matricial puede facilitar los cálculos matemáticos. Si se tienen k variables independientes x_1, x_2, \dots, x_k y n observaciones y_1, y_2, \dots, y_n , cada una de las cuales se puede expresar por la ecuación:

$$y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \dots + \beta_k X_{ki} + \varepsilon_i$$

En general, este modelo representa n ecuaciones que describen como se generan los valores de respuesta en el proceso científico. Con la notación matricial se pueden escribir las ecuaciones:

$$y = X\beta + \varepsilon;$$

donde:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \end{pmatrix} \quad X = \begin{pmatrix} 1 & x_{11} & \dots & x_{21} & \dots & x_{k1} \\ 1 & x_{12} & \dots & x_{22} & \dots & x_{k2} \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ 1 & x_{1n} & \dots & x_{2n} & \dots & x_{kn} \end{pmatrix} \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdot \\ \beta_k \end{pmatrix}$$

Entonces, la solución de mínimos cuadrados para la estimación de β presentada en el punto anterior involucra encontrar b para lo cual se minimiza:

$$SSE = (y - Xb)'(y - Xb)$$

El resultado se reduce a la solución de b en:

$$(X'X)b = X'y$$

Resolviendo la ecuación anterior se puede obtener la ecuación de predicción o ecuación de regresión al resolver un conjunto de k+1 ecuaciones en un número igual de incógnitas.

3. Propiedades de los estimadores de mínimos cuadrados: Las medias y las varianzas de los estimadores B_0, B_1, \dots, B_k se obtienen bajo determinadas suposiciones acerca de los errores aleatorios E_1, E_2, \dots, E_n que son iguales al caso de regresión lineal simple. Suponga que se consideran estos errores como independientes, cada uno con media cero y varianza σ^2 . Se puede demostrar que B_0, B_1, \dots, B_k son estimadores de los coeficientes de regresión $\beta_0, \beta_1, \dots, \beta_k$. Así mismo la varianza de las B's se obtienen gracias a los elementos de la inversa de la matriz A.

Se notará que los elementos fuera de la diagonal de $A=X'X$ representan las sumas de productos de elementos en las columnas de X , mientras que los elementos de la diagonal de A representan las sumas de cuadrados de elementos en las columnas de X . La matriz inversa $A^{-1}\sigma^2$ representan las varianzas de B_0, B_1, \dots, B_k sobre la diagonal principal y las covarianzas fuera de la diagonal. Las estimaciones de la varianza y de ahí los errores estándar se pueden calcular a partir del siguiente teorema:

Para la ecuación de regresión lineal

$$y = X\beta + \varepsilon$$

una estimación insesgada de σ^2 está dada por el cuadrado medio del error o cuadrado medio residual

$$s^2 = SSE / n - k - 1$$

donde;

$$SSE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Este teorema representa una generalización del definido para el caso de regresión lineal simple. Así mismo, s^2 es una medida de la variación de los errores de predicción o residuales.

La notación previa para la suma total de cuadrados es:

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2$$

La notación previa para la suma de cuadrados de regresión es:

$$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

Así, existen k grados de libertad asociados con SSR y $n-1$ grados de libertad para SST .

La utilidad del cuadrado medio residual es que se usa en la hipótesis y estimación de intervalos de confianza. Por otra parte el cuadrado medio del error es importante cuando se está buscando lo mejor de un conjunto de modelos en competencia.

4. Inferencias en regresión lineal múltiple: Una de las inferencias más útiles que se pueden realizar respecto a la calidad de la respuesta pronosticada y_0 correspondiente a los valores $x_{10}, x_{20}, \dots, x_{k0}$ es el intervalo de confianza para la respuesta promedio $\mu_{Y| x_{10}, x_{20}, \dots, x_{k0}}$. Si se supone que los errores son independientes y tienen distribución normal, entonces las B_j 's son normales con media, varianza y covarianza, y de aquí que:

$$\hat{Y}_0 = B_0 + \sum_{i=1}^n B_i x_{i0};$$

también tenga distribución normal y sea un estimador para la respuesta media sobre la cual se está intentando asociar los intervalos de confianza. La varianza de \hat{Y}_0 escrita en notación matricial como una función de $\sigma^2(X'X)^{-1}$, y el vector de condición x_0' es:

$$\sigma_{Y_0}^2 = \sigma^2 x_0'(X'X)^{-1} x_0;$$

después de reemplazar σ^2 por s^2 se puede determinar el intervalo del 100(1- α)% de confianza de $\mu_{Y| x_{10}, x_{20}, \dots, x_{k0}}$ del estadístico:

$$T = \frac{\hat{Y}_0 - \mu_{Y| x_{10}, x_{20}, \dots, x_{k0}}}{s \sqrt{x_0'(X'X)^{-1} x_0}};$$

la cual tiene una distribución t con n-k-1 grados de libertad.

Así el intervalo de confianza para $\mu_{Y| x_{10}, x_{20}, \dots, x_{k0}}$ con (1- α)100% para la respuesta media $\mu_{Y| x_{10}, x_{20}, \dots, x_{k0}}$ es:

$$\hat{Y}_0 - t_{\alpha/2} s \sqrt{x_0'(X'X)^{-1} x_0} < \mu_{Y| x_{10}, x_{20}, \dots, x_{k0}} < \hat{Y}_0 + t_{\alpha/2} s \sqrt{x_0'(X'X)^{-1} x_0}$$

la cantidad $s \sqrt{x_0'(X'X)^{-1} x_0}$ frecuentemente recibe el nombre de error estándar de predicción.

5. Selección de un modelo de predicción utilizando prueba de hipótesis: En muchas situaciones de regresión, los coeficientes individuales son importantes para el experimento. En otros casos es de mayor importancia la función completa para pronosticar la respuesta real en el rango de las variables consideradas. En este caso se pondría más énfasis en intervalos de confianza de la respuesta media y tal vez se restaría importancia a las inferencias de parámetros individuales.

En el experimento es probable que se utilice el análisis de regresión cuando se está interesado en la eliminación de variables si la situación lo permite, además que se puede llegar a una ecuación de predicción más fácil de trabajar.

Un criterio que en general se utiliza para ejemplificar lo adecuado de un modelo de regresión ajustado es el coeficiente de determinación múltiple:

$$R^2 = SSR/SST;$$

el cual indica que proporción de la variación total de la respuesta Y se aplica mediante el modelo ajustado. La raíz cuadrada de R^2 recibe el nombre de coeficiente de correlación múltiple entre Y y el conjunto X_1, X_2, \dots, X_k

Con frecuencia se expresa R^2 por 100% para interpretar en porcentaje la variación explicada por el modelo postulado.

6. Pruebas de subconjuntos de coeficientes y coeficientes individuales: La inclusión de cualquier variable sencilla en un sistema de regresión incrementará la suma de cuadrados de la regresión y de esta manera reducirá la suma de cuadrados del error. En consecuencia, se debe decidir si el incremento en la regresión es suficiente para garantizar su uso en el modelo. El uso de variables no importantes puede reducir la efectividad de la ecuación de predicción al incrementar la varianza de la respuesta estimada. Se puede probar que:

H₀: $\beta = 0$ (la variable no aporta información al modelo, puede ser eliminada).

H₁: $\beta \neq 0$ (la variable aporta información al modelo, puede ser eliminada).

Se debe calcular:

$$t = \frac{b_j - \beta_{j0}}{s\sqrt{C_{jj}}}$$

no se rechaza H_0 si

$$-t_{\alpha/2} < t < t_{\alpha/2};$$

donde $t_{\alpha/2}$ tiene $n-k-1$ grados de libertad.

7. Estudio de residuales: Los residuales o errores en el ajuste de regresión, con frecuencia son portadores de información de gran utilidad para el análisis de los datos. Los $e_i = y_i - \hat{y}_i$ $i=1,2,\dots,n$, son la contraparte numérica de los ε_i o errores del modelo, los cuales aclaran los puntos de datos sospechosos.

A través de la investigación el autor ha demarcado el marco teórico antes descrito como apoyo al desarrollo del proyecto.

Sistema de Variables

En base a la teoría anterior se determinan las variables Dependientes e Independientes.

En la tabla 2 podemos observar las variables independientes: parámetros (jitter, delay y pérdida de paquetes). La variable dependiente se define como QoS y va a permitir garantizar los requisitos mínimos para asegurar el nivel de servicio deseado.

Este modelo en estudio determinará la QoS a partir de las variables independientes jitter, delay y pérdida de paquetes.

La variable dependiente QoS y las independientes denominadas jitter y delay se cuantificaron en los siguientes cinco niveles: Malo, Regular, Bueno, Muy Bueno y Excelente y la Pérdida de paquetes en Malo y Excelente, partiendo de que el estado ideal de las comunicaciones es que no exista ningún tipo de retardo ni pérdidas de paquetes.

Tabla 2
Operacionalización de las Variables

Variables	Dimensiones	Indicadores
Parámetros	Jitter	$0 \text{ ms} \leq \text{Jitter} < 10 \text{ ms}$ (Excelente) $10 \text{ ms} \leq \text{Jitter} < 20 \text{ ms}$ (Muy Bueno) $20 \text{ ms} \leq \text{Jitter} < 30 \text{ ms}$ (Bueno) $30 \text{ ms} \leq \text{Jitter} < 40 \text{ ms}$ (Regular) $\text{Jitter} \geq 40 \text{ ms}$ (Malo)
	Delay	$0 \text{ ms} \leq \text{Delay} < 50 \text{ ms}$ (Excelente) $50 \text{ ms} \leq \text{Delay} < 100 \text{ ms}$ (Muy Bueno) $100 \text{ ms} \leq \text{Delay} < 150 \text{ ms}$ (Bueno) $150 \text{ ms} \leq \text{Delay} < 200 \text{ ms}$ (Regular) $\text{Delay} \geq 200 \text{ ms}$ (Malo)
	Perdida de Paquetes	$\text{PP} < 1\%$ (Excelente) $\text{PP} \geq 1\%$ (Malo)
Calidad de Servicio	QoS	Q=5 Excelente Q=4 Muy Bueno Q=3 Bueno Q=2 Regular Q=1 Malo

Fuente: El Autor (2007)

Definición de Términos

Ancho de Banda: Describe la capacidad de tasa de bits por unidad de tiempo de un medio determinado, de un protocolo o una conexión.

Calidad de Servicio (QoS): Es la medida de la calidad de transmisión y de la disponibilidad de una red o grupo de redes.

CoS (Clase de Servicio): Permite al administrador de la red agrupar diferentes flujos de paquetes, teniendo cada uno requisitos de latencia y ancho de banda diferentes.

Disponibilidad: Tiempo disponible para enrutar paquetes.

Diffserv (Diferenciación de Servicios): Asigna Clases de Servicio o niveles de prioridad a los paquetes mediante un proceso de marcado en el encabezado del paquete.

G.711: Técnica de codificación de la voz PCM a 64 Kbps. Con este formato se realiza la interconexión con las centrales públicas o entre enlaces de PBX.

G.726: Codificación ADPCM (Adaptive Differential Pulse Code Modulation) a 40, 32.24 y 16 Kbps. También se puede realizar la interconexión con centrales públicas o privadas si estas últimas soportan esta norma de codificación.

G.728: Describe la variación de bajo retardo de 16 Kbps de una compresión CELP (Code Excited Linear **Prediction Compression**).

G.729 y G.729a: Describe la compresión CELP que permite que la voz sea codificada a 8 Kbps.

G.731: Se utiliza para comprimir voz u otra señal de audio de servicios multimedia de baja velocidad como parte de la familia del estándar H.324.

Gatekeeper: Es un punto central de control en una red H.323, proporcionando servicios de control de llamada, traducción de direcciones y control de admisión. Además facilita el control del ancho de banda utilizado y localiza los distintos gateways cuando se necesita.

Gateway: Es el elemento encargado de hacer de puente entre la red telefónica convencional (RTB) y la red IP.

Jitter: Variación en los tiempos de llegada entre los paquetes.

Latencia o Retardo: Es el tiempo de transmisión para un paquete de datos que le toma en ir de un punto a otro en ambos sentidos (ida y vuelta).

Media Gateway: Se refiere a varios productos agrupados bajo el protocolo MGCP (Media Gateway Control Protocol). La principal misión de un Media Gateway es la conversión IP/TDM bajo el control de un Softswitch.

Megaco (Media Gateway Control): Es un protocolo de VoIP, que combina MGCP e IPDC. Es más sencillo que H.323.

MGCP (Media Gateway Controller Protocol): Protocolo de control de dispositivos, donde un gateway esclavo (MG, Media Gateway) es controlado por un maestro (MGC, Media Gateway Controller).

MOS (Mean Opinion Score): Asigna un valor a la calidad de la llamada en toda la red desde 1 (mala) a 5 (excelente). El valor de MOS real ha sido determinado en un ejercicio estadístico, un gran número de personas escuchando la misma llamada y valorándola de 1 a 5. Los analizadores pueden valorar marcaciones MOS por medio de algoritmos. Pueden darse datos MOS total o por llamadas.

PBX (Private Branch Exchange): Central telefónica utilizado en compañías para manejar llamadas externas e internas. La ventaja es que la compañía no necesita una línea telefónica para cada uno de sus teléfonos. Además las llamadas internas no salen al exterior y por tanto no son facturadas.

Pérdida de Paquetes (Packet Loss): Pérdida de alguna de las unidades de información, o paquetes, que componen un mensaje transmitido.

PESQ y PAMS: Fueron diseñados para aumentar el rango que cubría las medidas PSQM, incluye distorsión, filtrado y otras desigualdades del canal. Pero tampoco analizan todos los factores.

PSQM (Perceptual Speech Quality Measurement): Las medidas PSQM se realizan transmitiendo una señal conocida, analizándola después del CODEC en el otro extremo, se graba, se compara con la original y, de este modo, se obtiene un valor PSQM. Sin embargo, estas medida fueron diseñadas para analizar sólo los efectos de la compresión/ descompresión de las funciones del CODEC, es decir, PSQM no tiene capacidad de analizar los efectos causados por el trayecto a través de la red, como pueden ser la pérdida de paquetes o el jitter.

Retardo de Transmisión: Tiempo que tarda un dispositivo en enviar el paquete completo a través de la interfaz de salida.

Retardo de Propagación: Tiempo que tarda un bit desde el transmisor hasta el enlace del receptor.

Retardo de Conmutación: Tiempo que toma para un dispositivo el transmitir un paquete después de recibirlo de otra fuente.

SLA (Service Level Agreement): Contrato que especifica los parámetros de QoS entre el proveedor y el cliente.

SS7 (Sistema de canal común de señalización N^a 7): Es un estándar global para telecomunicaciones definido por la Unión Internacional de Telecomunicaciones. Define los procedimientos y protocolos mediante los cuales los elementos de la Red Telefónica Conmutada intercambian información sobre una red de señalización digital para establecer, enrutar, facturar y controlar llamadas, tanto a terminales fijos como móviles.

ToS (Tipo de Servicio): Campo en la cabecera de Protocolo de Internet (IP) que permite que tenga lugar una clase de servicio.

CAPITULO III

MARCO METODOLOGICO

Tipo de Investigación

El presente estudio propone un modelo para medir QoS en la red de telefonía digital de Intercable, de acuerdo a los objetivos planteados, se incorpora en el tipo de investigación de campo siguiendo la modalidad de un proyecto especial ya que permite no solo observar, sino recolectar los datos directamente de la realidad para posteriormente analizar e interpretar los resultados.

De acuerdo con la definición presentada por el Manual de la Universidad Centroccidental Lisandro Alvarado (2002), se entiende por Investigación de Campo “la aplicación del método científico en el tratamiento de un sistema de variables y sus relaciones, las cuales conducen a conclusiones y al enriquecimiento de un campo del conocimiento o disciplina inherente a la especialidad, con la sustentación de los experimentos y observaciones realizadas” (p.2).

Este estudio se sitúa en el ámbito no experimental descriptivo donde se han definido un conjunto de variables para describirlas y analizarlas en determinado momento. En relación a los estudios descriptivos según Balestrini (2001),

“Este diseño, permite establecer una relación entre los objetivos y la realidad de la situación de campo, observar y recolectar los datos directamente de la realidad, en su situación natural; profundizar en la comprensión de los hallazgos encontrados con la aplicación de los instrumentos; y proporcionarle al investigador una lectura de la realidad objeto de estudio más rica en cuanto al conocimiento de la misma, para plantear hipótesis futuras en otros niveles de investigación.” (p.132).

El Manual de presentación de trabajos de grado y tesis doctoral de la Universidad Pedagógica Experimental Libertador (2004), define como proyecto especial a:

“Trabajos que llevan a creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados, o que respondan a necesidades e intereses de tipo cultural. Se incluyen en esta categoría los trabajos de elaboración de libros de texto y de materiales de apoyo educativo, el desarrollo de software, prototipos y de productos tecnológicos en general, así como también los de creación literaria y artística” (p.8).

En este sentido, se estudió un modelo que permita determinar la QoS en los usuarios de la red de telefonía digital de Intercable y sirva como herramienta para detectar fallas y monitorear los puntos telefónicos.

Población

Según Balestrini (2001): “ Desde el punto de vista estadístico, una población o universo puede estar referido a cualquier conjunto de elementos de las cuales pretendemos indagar y conocer sus características, o una de ellas, y para el cual serán válidas las conclusiones obtenidas en la investigación” (p. 137).

La población comprende las llamadas que se realizan desde la red de telefonía digital de Intercable por un período de cinco días comprendido desde el lunes 16 de Julio del 2007 al viernes 20 de Julio del 2007.

Muestra

Balestrini (2001), define: “La muestra estadística es una parte de la población, o sea, un número de individuos u objetos seleccionados científicamente, cada uno de los cuales es un elemento del universo” (p. 141).

En este estudio se tomó como muestra 50 observaciones de las llamadas realizadas desde la red de telefonía digital de Intercable las cuales fueron

monitoreadas por un período de cinco días hábiles, comprendido desde el lunes 16 de Julio de 2007 hasta el Viernes 20 de Julio de 2007, en un horario comprendido de 10:00 a.m. a 12:00 p.m.

La muestra es representativa para la investigación, debido a que estadísticamente 50 observaciones son suficientes para un análisis de regresión lineal múltiple, como lo expresa Myer (2004), el cual afirma que n debe ser mayor que k , siendo n y k el número de muestras y de variables independientes respectivamente.

Fases del Estudio

Seguidamente se explica cada una de las diferentes fases requeridas para el desarrollo de la presente investigación:

Fase 1. Diagnóstico

Esta fase consistió en determinar las necesidades presentadas en la red de telefonía digital de Intercable, para así determinar que no se cuenta con un monitor de QoS, por lo que se propone en esta investigación desarrollar un modelo que a partir de los parámetros dependiente de la misma pueda predecirla a través de una herramienta automatizada adaptable a los cambios que pueda sufrir la red.

Fase 2. Revisión Bibliográfica

En esta fase se realiza una búsqueda de información en fuentes bibliográficas, documentales y electrónicas que permita desarrollar la parte teórica así como determinar los aspectos que conducirán al desarrollo de los objetivos de la investigación y a la vez determinar los parámetros dependientes de la QoS.

Fase 3. Construcción del Modelo

A partir de la investigación realizada se construyó un modelo de predicción de calidad de servicio determinando que el método de regresión lineal simple se adapta al objetivo de la investigación de acuerdo a los datos recolectados.

Fase 4. Evaluación del Modelo

Se realizó una simulación del modelo de regresión lineal con la información capturada de cada parámetro dependiente de la QoS a través de lecturas SNMP, para luego chequear los resultados generados de manera tal que permita determinar el estado de la red y de las llamadas.

CAPITULO IV

Análisis de Resultados

Construcción del Modelo

Esta fase respondió al tercer (3er) objetivo específico de la investigación, el cual fue: Construir un modelo de predicción de QoS en función del análisis de los parámetros jitter, delay y pérdida de paquetes.

Para la generación del modelo que representa la QoS partiendo que se necesita estudiar más de dos variables se utilizó la técnica de Análisis de Regresión Lineal Múltiple, en el que se relaciona la variable QoS que se desea explicar, con los parámetros dependientes de la misma, definidos como jitter, delay y pérdida de paquetes.

En primera instancia se procedió a monitorear por un período de 5 días hábiles desde las 10 a.m. hasta las 12 p.m. los parámetros dependientes de la QoS: Jitter, delay y pérdida de paquetes a través de lecturas SNMP (Anexo A). Las muestras fueron tomadas en el puerto ethernet del router principal o CMTS BSR 64000.

Siguiendo los valores estándares de la industria, el autor cuantifica el jitter, delay y la pérdida de paquetes como se muestra a continuación:

$0 \text{ ms} \leq \text{Delay} < 50 \text{ ms}$ (Excelente)
 $50 \text{ ms} \leq \text{Delay} < 100 \text{ ms}$ (Muy Bueno)
 $100 \text{ ms} \leq \text{Delay} < 150 \text{ ms}$ (Bueno)
 $150 \text{ ms} \leq \text{Delay} < 200 \text{ ms}$ (Regular)
 $\text{Delay} \geq 200 \text{ ms}$ (Malo)

$0 \text{ ms} \leq \text{Jitter} < 10 \text{ ms}$ (Excelente)
 $10 \text{ ms} \leq \text{Jitter} < 20 \text{ ms}$ (Muy Bueno)
 $20 \text{ ms} \leq \text{Jitter} < 30 \text{ ms}$ (Bueno)
 $30 \text{ ms} \leq \text{Jitter} < 40 \text{ ms}$ (Regular)
 $\text{Jitter} \geq 40 \text{ ms}$ (Malo)

y,

Pérdida de paquetes $< 1\%$ (Excelente)

Perdida de paquetes $\geq 1\%$ (Malo)

Para demostrar el modelo de regresión lineal por criterio del investigador se cuantifica Q con los siguientes valores:

Q=5 Excelente

Q= 4 Muy Bueno

Q= 3 Bueno

Q= 2 Regular

Q= 1 Malo

A continuación se muestran los valores capturados y la QoS cuantificada, por ejemplo en la muestra número 22 del día Miércoles tenemos que:

Muestra N ^a	X1(Delay)	X2(Jitter)	X3(PP)	X4(QoS)
22	90	20	5	1

Según los niveles descritos anteriormente el delay se encuentra en el rango Muy Bueno, el jitter en Muy Bueno pero la pérdida de paquetes se encuentra en Malo, tomando para la cuantificación de QoS el peor nivel que muestre alguna de las tres variables. En este caso sería QoS= 1(Malo). Se tomo este criterio porque lo ideal sería que no hubiesen retardos en las comunicaciones ni se perdieran paquetes para garantizar un 100% la QoS. La misma cuantificación se aplica al resto de la muestra que se ilustra a continuación:

CAPTURA DE PAQUETES DEL DIA LUNES

Tabla 3

Data capturada el día lunes 16-07-07

Muestra N ^a	X1(Delay)	X2(Jitter)	X3(PP)	X4(QoS)
1	42	3	0	5
2	75	11	0	4
3	45	5	0	5
4	80	10	0	4
5	85	8	0	4
6	48	4	0	5

7	40	6	0	5
8	87	9	0	4
9	46	7	0	5
10	40	3	0	5

Fuente: Lectura SNMP en el router principal (2007)

CAPTURA DE PAQUETES DEL DIA MARTES

Tabla 4

Data capturada el día martes 17-07-07

Muestra N ^a	X1(Delay)	X2(Jitter)	X3(PP)	X4(QoS)
11	1352	3	0	1
12	0	20	0	3
13	0	0	0	5
14	0	112	0	1
15	0	3	0	5
16	0	0	0	5
17	0	4	0	5
18	0	3	0	5
19	0	4	0	5
20	0	118	0	1

Fuente: Lectura SNMP en el router principal (2007)

CAPTURA DE PAQUETES DEL DIA MIERCOLES

Tabla 5

Data capturada el día miércoles 18-07-07

Muestra N ^a	X1(Delay)	X2(Jitter)	X3(PP)	X4(QoS)
21	0	20	1	1
22	90	20	5	1
23	1225	20	0	1
24	860	7	0	1
25	0	118	0	1
26	0	25	0	3
27	0	4	0	5
28	20	5	0	5
29	0	7	0	5
30	87	20	0	3

Fuente: Lectura SNMP en el router principal (2007)

CAPTURA DE PAQUETES DEL DIA JUEVES

Tabla 6

Data capturada el día jueves 19-07-07

Muestra N ^a	X1(Delay)	X2(Jitter)	X3(PP)	X4(QoS)
31	46	0	1	1
32	75	10	5	1
33	1300	3	0	1
34	0	112	0	1
35	42	8	0	5
36	120	3	0	3
37	28	3	0	5
38	35	4	0	5
39	0	0	0	5
40	0	7	2	1

Fuente: Lectura SNMP en el router principal (2007)

CAPTURA DE PAQUETES DEL DIA VIERNES

Tabla 7

Data capturada el día viernes 20-07-07

Muestra N ^a	X1(Delay)	X2(Jitter)	X3(PP)	X4(QoS)
41	0	5	1	1
42	0	10	0	4
43	40	8	0	5
44	75	20	0	3
45	63	3	0	4
46	110	5	0	3
47	26	7	0	5
48	15	10	0	4
49	18	7	0	5
50	1150	8	0	1

Fuente: Lectura SNMP en el router principal (2007)

Teniendo los datos que sirven como base para determinar el modelo que más se ajusta a la investigación se procede a introducirlos en el programa de análisis estadístico SPSS V11.5. En el Anexo B podemos observar los resultados arrojados por el software al analizar los datos a través de regresión lineal múltiple y que a continuación serán demostrados.

En base a la teoría expuesta en el marco teórico basada en el texto de Myers(2004) definimos el modelo de regresión lineal múltiple ya que existe más de una variable independiente, donde:

k: Numero de variables independientes.

x_1, x_2, \dots, x_k : Variables independientes.

Media de Y: $Y | x_1, x_2, \dots, x_k \rightarrow \mu_{Y | x_1, x_2, \dots, x_k} = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$

B_i: Coeficientes de regresión.

Respuesta estimada o ecuación de regresión muestral:

$$y = b_0 + b_1 x_1 + \dots + b_k x_k$$

1. Se debe encontrar los coeficientes de regresión

$k=3$; $n=50$; $n > k$ $50 > 3$; $i=1, 2, 3, \dots, 50$

Se generan $k + 1 = 4$ ecuaciones normales:

$$50b_0 + b_1 \sum_{i=1}^{50} x_{1i} + b_2 \sum_{i=1}^{50} x_{2i} + b_3 \sum_{i=1}^{50} x_{3i} = \sum_{i=1}^{50} y_i$$

$$b_0 x_{1i} + b_1 \sum_{i=1}^{50} x_{1i}^2 + b_2 \sum_{i=1}^{50} x_{1i} x_{2i} + b_3 \sum_{i=1}^{50} x_{1i} x_{3i} = \sum_{i=1}^{50} x_{1i} y_i$$

$$b_0 x_{2i} + b_1 \sum_{i=1}^{50} x_{2i} x_{1i} + b_2 \sum_{i=1}^{50} x_{2i}^2 + b_3 \sum_{i=1}^{50} x_{2i} x_{3i} = \sum_{i=1}^{50} x_{2i} y_i$$

$$b_0 x_{3i} + b_1 \sum_{i=1}^{50} x_{3i} x_{1i} + b_2 \sum_{i=1}^{50} x_{3i} x_{2i} + b_3 \sum_{i=1}^{50} x_{3i}^2 = \sum_{i=1}^{50} x_{3i} y_i$$

Resolviendo las sumatorias resulta:

$$\sum_{i=1}^{50} x_{1i} = 7365; \quad \sum_{i=1}^{50} x_{2i} = 812; \quad \sum_{i=1}^{50} x_{3i} = 9; \quad \sum_{i=1}^{50} y_i = 193$$

$$\sum_{i=1}^{50} x_{1i}^2 = 7177259 \quad \sum_{i=1}^{50} x_{1i} x_{2i} = 60316 \quad \sum_{i=1}^{50} x_{1i} x_{3i} = 450 \quad \sum_{i=1}^{50} x_{1i} y_i = 11830$$

$$\sum_{i=1}^{50} x_{2i}^2 = 57410 \quad \sum_{i=1}^{50} x_{2i} x_{3i} = 139 \quad \sum_{i=1}^{50} x_{2i} y_i = 1759$$

$$\sum_{i=1}^{50} x_{3i}^2 = 31 \quad \sum_{i=1}^{50} x_{3i} y_i = 24$$

Sustituyendo en la matriz:

$$\begin{aligned} 50b_0 + 7365b_1 + 812b_2 + 9b_3 &= 193 \\ 7365b_0 + 7177259b_1 + 60316b_2 + 450b_3 &= 11830 \\ 812b_0 + 60316b_1 + 57410b_2 + 139b_3 &= 1759 \\ 9b_0 + 450b_1 + 139b_2 + 31b_3 &= 24 \end{aligned}$$

Con apoyo en el texto de Duane(1996), titulado “*Matlab edición de estudiante*”

y usando el software Matlab7 el sistema de ecuaciones anterior da como resultado:

$$b_0 = 4.9807 \quad b_1 = -0.0031 \quad b_2 = -0.0354 \quad b_3 = -0.4677;$$

lo que coincide con los coeficientes calculados por el SPSS y define la ecuación de regresión muestral como:

$$\hat{y} = 4.9807 - 0.0031x_1 - 0.0354 x_2 - 0.4677x_3$$

Ahora vamos a estimar QoS a partir de la ecuación con la muestra número 8 donde el delay=87, jitter=9 y perdida de paquetes= 0

$$\hat{y} = 4.9807 - 0.0031(87) - 0.0354 (9) - 0.4677(0) = 4.3924 = 4 \text{ (Muy Bueno)}$$

2. Se estima la varianza σ^2 para medir la variación de los errores y determinar los grados de libertad.

$$\sigma^2 = s^2 = \frac{SSE}{n-k-1} \quad ; \text{ con } n=50 \text{ y } k=3$$

donde;

$$SSE = SST - SSR \quad y;$$

$$SST = \frac{\sum_{i=1}^{50} y_i^2 - \left(\frac{\sum_{i=1}^{50} y_i}{n} \right)^2}{n}$$

$$SSR = \frac{\sum_{i=1}^{50} b_j g_j - \left(\frac{\sum_{i=1}^{50} y_i}{n} \right)^2}{n}$$

$$\text{Resolviendo: } \left(\frac{\sum_{i=1}^{50} y_i}{n} \right)^2 = 37249 \quad ; \quad \sum_{i=1}^{50} y_i^2 = 865$$

$$SST = 120.02 \text{ con } n-1 = 49 \text{ grados de libertad}$$

Resolviendo con Matlab $\sum_{i=1}^{50} b_j g_j = 521.0517$, entonces

$$SSR = 106.1287 \text{ con } k=3 \text{ grados de libertad;}$$

$$SSE = 13.8913 \text{ con } n-k-1=46 \text{ grados de libertad;}$$

$$\sigma^2 = s^2 = 0.3019$$

3. Se calcula los intervalos de confianza para la respuesta media

Sustituyendo la muestra nº 44 en la ecuación de regresión muestral:

$\hat{y} = 4.9807 - 0.0031(75) - 0.0354(20) - 0.4677(0) = 4.0402$; lo que representa una respuesta estimada para la muestra 44.

Los intervalos de confianza vienen representados por:

$$\hat{Y}_0 - t_{\alpha/2} S \sqrt{x_0'(X'X)^{-1}x_0} < \mu_{Y|X_0, X_1, X_2, \dots, X_K} < \hat{Y}_0 + t_{\alpha/2} S \sqrt{x_0'(X'X)^{-1}x_0}; \quad (1)$$

donde:

$$x_0 = \begin{pmatrix} 1 \\ 75 \\ 20 \\ 0 \end{pmatrix} \quad x_0' = [1 \quad 75 \quad 20 \quad 0]$$

$$x = \begin{pmatrix} 50 & 7365 & 812 & 9 \\ 7365 & 7177259 & 60316 & 450 \\ 812 & 60316 & 57410 & 139 \\ 9 & 450 & 139 & 31 \end{pmatrix}$$

$$x' = \begin{pmatrix} 50 & 7365 & 812 & 9 \\ 7365 & 7177259 & 60316 & 450 \\ 812 & 60316 & 57410 & 139 \\ 9 & 450 & 139 & 31 \end{pmatrix}$$

$$x'x = \begin{pmatrix} 0 & 0.0053 & 0 & 0 \\ 0.0053 & 5.1517 & 0.0436 & 0.0003 \\ 0 & 0.0436 & 0.0007 & 0 \\ 0 & 0.0003 & 0 & 0 \end{pmatrix} \times 10^{-13}$$

$$(x'x)^{-1} = \begin{pmatrix} 0.0011 & 0 & 0 & -0.0005 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.0005 & 0 & 0 & 0.0012 \end{pmatrix}$$

Por lo tanto: $S^2 = 0.3019$; $S = 0.5495$

Para un intervalo de confianza del 95% el porcentaje promedio para la muestra n° 44 sería:

$$(1-\alpha)100\% = 95\%$$

$$\alpha = 0.05$$

$$\alpha/2 = 0.025 \text{ con } n-k-1=46 \text{ grados de libertad.}$$

Por la tabla de distribución t (Anexo C) $t_{\alpha/2} = 1.960$

Sustituyendo en (1) resultaría:

$$4.0156 < MY|_{x_{10}, x_{20}, x_{30}} < 4.0648$$

Por lo que el valor estimado anteriormente es aceptable ya que esta entre estos límites.

4. Ahora se calcula un intervalo de predicción para una respuesta individual Y_0

Sustituyendo en:

$$\hat{Y}_0 - t_{\alpha/2} s \sqrt{1-x_0'(X'X)^{-1}x_0} < Y_0 < \hat{Y}_0 + t_{\alpha/2} s \sqrt{1+x_0'(X'X)^{-1}x_0};$$

$$\text{Resolviendo } 2,9629 < Y_0 < 5.1175$$

5. Calculando la varianza de los coeficientes

$$\sigma^2\beta_i = C_{ii} \sigma^2 ; \text{ donde}$$

$$C_{ii} = (x'x)^{-1} = \begin{pmatrix} C_{00} & C_{01} & C_{02} & C_{03} \\ C_{10} & C_{11} & C_{12} & C_{13} \\ C_{20} & C_{21} & C_{22} & C_{23} \\ C_{30} & C_{31} & C_{32} & C_{33} \end{pmatrix}$$

$$\sigma^2\beta_0 = C_{00} \sigma^2 = 0.000332$$

$$\sigma^2\beta_1 = C_{11} \sigma^2 = 0.000362$$

$$\sigma^2\beta_2 = C_{22} \sigma^2 = 0.000694$$

$$\sigma^2\beta_3 = C_{33} \sigma^2 = 0.000362$$

6. Para la selección del modelo de predicción usando la prueba de la hipótesis:

$R^2 = SSR / SST = 0.8843$ que representa el 88.43 %, lo cual significa que 88,43% de la variación de la QoS se explican mediante el modelo calculado de regresión lineal.

Una forma de explicar que la regresión demostrada por el modelo es significativa es la distribución F expresada como :

$$F = SSR/ks^2 = 117.1786 \quad ;$$

$$F_{\alpha}(k, n-k-1) = (3,46) \text{ con } \alpha = 0.05$$

Por la tabla de distribución de F(Anexo D):

$$F_g = 2.816$$

Como $F > F_g$ se puede decir que la ecuación de regresión es significativa, sin embargo, no se excluye la posibilidad de que:

a. El modelo de regresión lineal desarrollado no es el único que puede utilizarse para explicar los datos.

b. Al incluir otras variables en el modelo podría haber sido mas eficaz , o tal vez con la eliminación de una o más variables en el modelo.

7. A continuación se demuestra la importancia de las variables en el modelo mediante la prueba de los coeficientes.

Existen dos hipótesis

$$H_0 = \beta_1 = 0 \text{ (x no aporta información al modelo)}$$

$$H_1 = \beta_1 \neq 0 \text{ (x aporta información al modelo)}$$

$$t = b_j - B_{j0} / s\sqrt{C_{jj}}$$

Para X₁ (Delay)

$$t = b_1 - B_{10} / s\sqrt{C_{11}}$$

Suponemos $B_{10}=0$ entonces,

$$t = -0.0031 - 0 / 0.5495 \sqrt{0.0012 * 10_{.12}}$$

$$t = -170097.3918$$

Como este valor está muy alejado de cero se desecha la H_0 demostrando que la variable X_1 (Delay) no se puede eliminar ya que aporta información en el modelo.

Para X₂ (Jitter)

$$t = b_1 - B_{10} / s\sqrt{C_{22}}$$

Suponemos $B_{10}=0$ entonces,

$$t = -0.0354 - 0 / 0.5495 \sqrt{0.0023 * 10_{.12}}$$

$$t = -117633.2454$$

Como este valor está muy alejado de cero se desecha la H_0 demostrando que la variable X_2 (Jitter) no se puede eliminar ya que aporta información en el modelo.

Para X_3 (Pérdida de Paquetes)

$$t = b_1 - B_{10} / s \sqrt{C_{33}}$$

Suponemos $B_{10}=0$ entonces,

$$t = -0.0031 - 0 / 0.5495 \sqrt{0.0012 * 10_{.12}}$$

$$t = -170097.3918$$

Como este valor está muy alejado de cero se desecha la H_0 demostrando que la variable X_3 (Pérdida de Paquetes) no se puede eliminar ya que aporta información en el modelo.

Como los t de cada variable son muchos menores que cero, esto implica que $Pt_k \ll \alpha = 0.05$ para todo $k = 1,2,3$, con lo cual se puede concluir una vez mas que existe suficiente evidencia que indica, que las tres variables predictoras x_1 , x_2 y x_3 aportan información significativa en el modelo, por lo tanto no se debe eliminar ninguna del modelo.

9. Ahora se realiza un estudio de los residuos o errores en el ajuste de regresión ya que representan información importante, los cuales se resumen a continuación:

Nº Muestra	y_i	\hat{Y}_i	$e = y_i - \hat{y}_i$
1	5	4,7443	0,2557
2	4	4,3588	-0,3588
3	5	4,6642	0,3358
4	4	4,3787	-0,3787
5	4	4,4340	-0,4340
6	5	4,6903	0,3097
7	5	4,6443	0,3557
8	4	4,3924	-0,3924
9	5	4,5903	0,4097
10	5	4,7505	0,2495

11	1	0,6833	0,3167
12	3	4,2727	-1,2727
13	5	4,9807	0,0193
14	1	1,0159	-0,0159
15	5	4,8745	0,1255
16	5	4,9807	0,0193
17	5	4,8391	0,1609
18	5	4,8745	0,1255
19	5	4,8391	0,1609
20	1	0,8035	0,1965
21	4	3,8050	0,1950
22	1	1,6552	-0,6552
23	1	0,4752	0,5248
24	1	2,0669	-1,0669
25	1	0,8035	0,1965
26	3	4,0957	-1,0957
27	5	4,8391	0,1609
28	5	4,7417	0,2583
29	5	4,7329	0,2671
30	4	4,0030	-0,0030
31	5	4,8381	0,1619
32	4	4,3942	-0,3942
33	1	0,8445	0,1555
34	1	1,0159	-0,0159
35	5	4,5673	0,4327
36	3	4,5025	-1,5025
37	5	4,7877	0,2123
38	5	4,7306	0,2694
39	5	4,9807	0,0193
40	5	3,7975	1,2025
41	5	4,3360	0,6640
42	5	4,6267	0,3733
43	5	4,5735	0,4265
44	4	4,0402	-0,0402
45	4	4,6792	-0,6792
46	3	4,4627	-1,4627
47	5	4,6523	0,3477
48	5	4,5802	0,4198
49	5	4,6771	0,3229
50	1	1,1325	-0,1325

La gráfica que ilustra los datos anteriores es la siguiente:

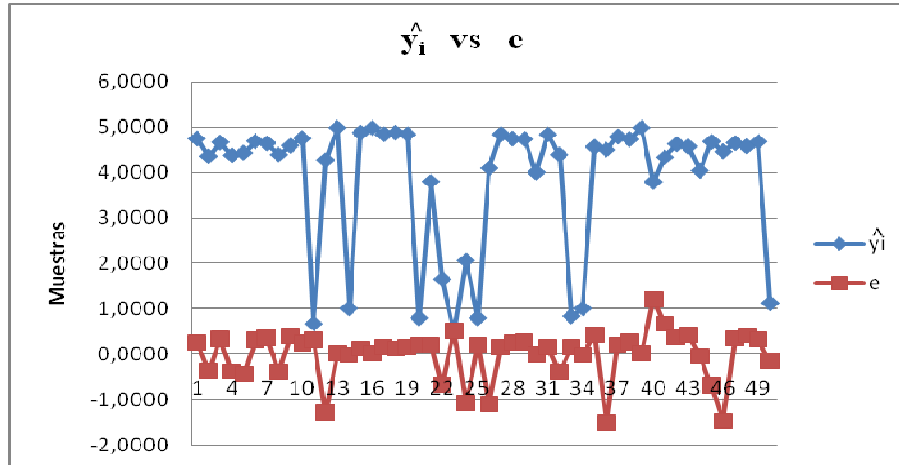


Figura 9. y_i calculada Vs el error.

La gráfica anterior se muestra ideal ya que el error en casi todos sus puntos da muy cercano a cero. Algunos se alejan un poco del eje cero pero no es muy significativo ya que son aquellos puntos que se encuentran en el 12% de error que representó el R^2 de la ecuación, pero se mantienen dentro de los intervalos de confianza.

Análisis y Comparación de Resultados

Con la realización de la construcción del modelo, basado en las 50 muestras tomadas, se generó el modelo,

$$\hat{y} = 4.9807 - 0.0031x_1 - 0.0354 x_2 - 0.4677x_3$$

que describe la QoS a través de la relación de sus parámetros delay, jitter y pérdida de paquetes.

A través del estudio anterior se demostró que el modelo encontrado, es apropiado y lo confirma aún más la relación R^2 la cual dio como resultado 0.8843, lo que quiere

decir que 88,43% de la variación de las muestras se explican mediante este modelo de regresión lineal. El intervalo de confianza resulto con un 95%.

La gráfica resultante obtenida de la y_i calculada muestra que las variables dependientes de la QoS, delay, jitter y perdida de paquetes fueron relacionadas mediante el modelo propuesto ya que se encuentran muy cerca de los valores de y_i , y otra manera de representarlo fue a través de la gráfica del error que resultó muy cercana a cero lo que afirma nuevamente que el modelo se adecúa al problema.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Luego de realizar el diagnóstico de la red de telefonía digital de Intercable a través de la observación directa del investigador se determinó la necesidad de buscar un método que permita obtener la QoS de las llamadas de una forma más rápida y amigable, ya que actualmente el monitoreo se realiza a través de las lecturas de tráfico obtenidas con el uso del protocolo SNMP las cuales deben ser traducidas para determinar los valores de los parámetros.

Por medio de la investigación bibliográfica se cumplió con el objetivo número dos de esta investigación, y se constató que los parámetros que miden la QoS son el jitter, el delay y la pérdida de paquetes, los cuales deben mantenerse dentro de los valores jitter < 30 ms., delay < 150 ms y pérdida de paquetes < 1%. De estos valores estándares aceptados por la industria, depende que las llamadas se escuchen con claridad, libres de eco, es decir, con una QoS óptima.

En la investigación desarrollada el objetivo número tres consistió en relacionar mediante una ecuación matemática las variables dependientes, descritas como jitter, delay y pérdida de paquetes para construir un modelo que permitió predecir la QoS.

Se demostró que el método de regresión lineal múltiple cumple con el objetivo propuesto, ya que permite estudiar más de dos variables como es el caso de este estudio, logrando definir el modelo que simula la QoS.

Después de calcular los coeficientes de regresión que generaron el modelo $y = 4.9807 - 0.0031x_1 - 0.0354 x_2 - 0.4677x_3$, se calculó la varianza que representa la variación de los errores la cual dió como resultado 0.3019, el cual es un estimador de la dispersión de los errores. Este valor indica que el modelo tiene una excelente

aproximación al fenómeno real pues este valor se encuentra muy cercano a cero, lo cual estadísticamente es muy pequeño.

Se determinaron los intervalos de confianza con un nivel de confiabilidad del 95% y los resultados obtenidos fueron aceptables, entre ellos se encuentran los valores calculados. También se determinó un intervalo de predicción que resulta un poco más amplio pero más propenso a errores, sin embargo da un estimado aceptable de cuanto puede variar la QoS.

También se demostró la relación del modelo con el caso de estudio al calcular el coeficiente de determinación múltiple R^2 que resultó 0.8843, entre la variable dependiente QoS y las independientes jitter, delay y pérdida de paquetes, el cual explica que el modelo representa un total de 88,43% de las 50 muestras monitoreadas. También se aplicó la distribución F y demostró que el modelo es significativo. A través de estos dos métodos utilizados se demostró la adaptación del modelo con la realidad estudiada.

A través de la prueba de las hipótesis de los coeficientes de las variables independientes jitter, delay y pérdida de paquetes, se pudo demostrar que las tres variables en estudio son indispensables para explicar el modelo $y = 4.9807 - 0.0031x_1 - 0.0354 x_2 - 0.4677x_3$, por lo que no se debe eliminar ninguna ya que todas aportan información significativa. Los valores probados de cada variable resultaron para el delay -170097.3918, para el jitter -117633.2454 y para la pérdida de paquetes -170097.3918 los cuales se encuentran muy alejados de cero. Según la teoría de regresión múltiple indica que el aporte de las variables al modelo son valores importantes al resultar muy alejadas de cero.

Para concluir con la demostración del modelo $y = 4.9807 - 0.0031x_1 - 0.0354 x_2 - 0.4677x_3$, se determinaron los errores residuales por medio de Q y Qcalculada que permiten obtener gráficas donde se representa la QoS y el error producido (ver Figura 9). Los resultados observados indicaron que ninguno de sus valores produce alarma porque se encuentra muy cercano a cero.

Para finalizar la investigación y cumplir con el último objetivo se evaluó el modelo propuesto a través de la simulación de la ecuación resultante en un software en visual basic 6.0, como se muestra en el capítulo siguiente, donde se puede observar la variación de la QoS versus el error, de acuerdo a los datos suministrados de las variables independientes jitter, delay y pérdida de paquetes.

El resultado del modelo y la clasificación de sus parámetros permiten representar cualquier muestra y dar una aproximación del nivel de QoS, lo que beneficia a los administradores de Intercable ya que pueden mantener esta variable dependiente monitoreada, proceso que al día de hoy se realiza de forma engorrosa.

El hecho de utilizar el método de regresión lineal múltiple para obtener este modelo de estimación de la QoS en una red de telefonía digital no excluye la posibilidad de estudiar otros métodos matemáticos, estadísticos, de algoritmos genéticos ó lógica difusa, que también se aproximen a una solución.

Recomendaciones

- 1.** Monitorear datos de otras empresas o instituciones similares y validar los resultados obtenidos por el modelo propuesto con el fin de realizar estudios de confiabilidad y aplicabilidad de la generalización del modelo.
- 2.** Incorporar nuevas variables al modelo que se consideren se relacionan con la QoS en VoIP con el propósito de elevar la confiabilidad y disminución de los márgenes de errores.
- 3.** Ensayar aproximaciones al modelo haciendo análisis de data usando enfoques basados en otros métodos matemáticos, algoritmos genéticos o lógica difusa.
- 4.** Construir un software que calcule en tiempo real la QoS. El mismo debe leer en tiempo real los valores del jitter, delay y pérdida de paquetes y calcular la QoS con el

modelo desarrollado. También debe mostrar la gráfica que representa la QoS y un conjunto de alarmas cuando esta este disminuyendo para alertar al administrador.

CAPITULO VI

SIMULACION DEL MODELO DE PREDICCIÓN DE CALIDAD DE SERVICIO PARA LA RED DE TELEFONIA DIGITAL DE INTERCABLE

Para cumplir con el último objetivo específico planteado en la investigación se realiza una simulación del modelo obtenido $y = 4.9807 - 0.0031x_1 - 0.0354x_2 - 0.4677x_3$, que permita evaluarlo y demostrar el comportamiento del mismo.

Para esto, se construye un software en Visual Basic 6.0 que al ejecutarse muestra la pantalla que se ilustra en la Figura 9.

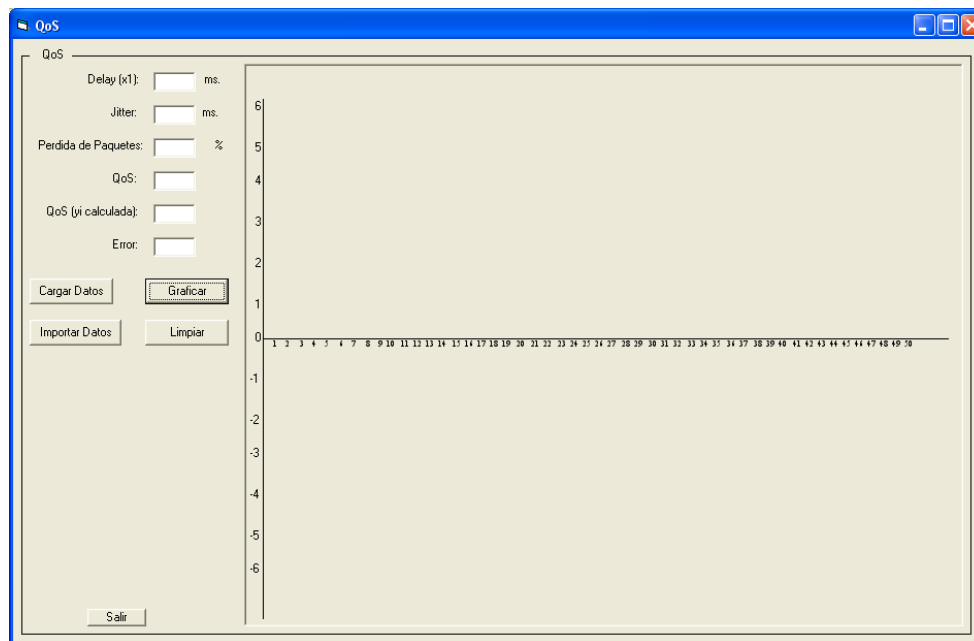


Figura 10. Pantalla principal del software de simulación

Se muestran los siguientes botones: Cargar Datos, Graficar e Importar Datos.

El primer botón Cargar Datos, permite introducir los datos para medir la QoS en una llamada, al pulsarlo nos muestra la pantalla que se muestra en la Figura 11 la cual espera por los valores de las variables jitter, delay, perdida de paquetes, QoS. Al pulsar aceptar va a mostrar estos datos en la pantalla principal.

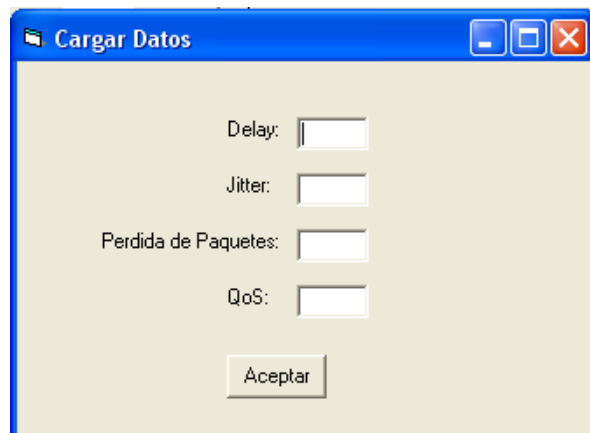


Figura 11. Pantalla para cargar los datos de una llamada

El software para determinar el valor de la QoS calculada introduce estos datos en la ecuación determinada por el Análisis de Regresión lineal el cual dio como resultado:

$$\hat{y} = 4.9807 - 0.0031x_1 - 0.0354 x_2 - 0.4677x_3$$

Luego de introducir los datos y realizar automáticamente el cálculo de la Q calculada, al pulsar el botón Graficar podemos observar la gráfica que define la QoS calculada y el error el cual viene dado por:

$$e = y_i - \hat{y}_i$$

Mediante este proceso podemos observar la QoS de cada llamada como se muestra en la Figura 12, y para fines de la simulación el error arrojado por el modelo el cual no es muy significativo y se demuestra al observar las gráficas generadas por el software que permite constatar que el modelo simulado refleja la realidad del comportamiento de la QoS.

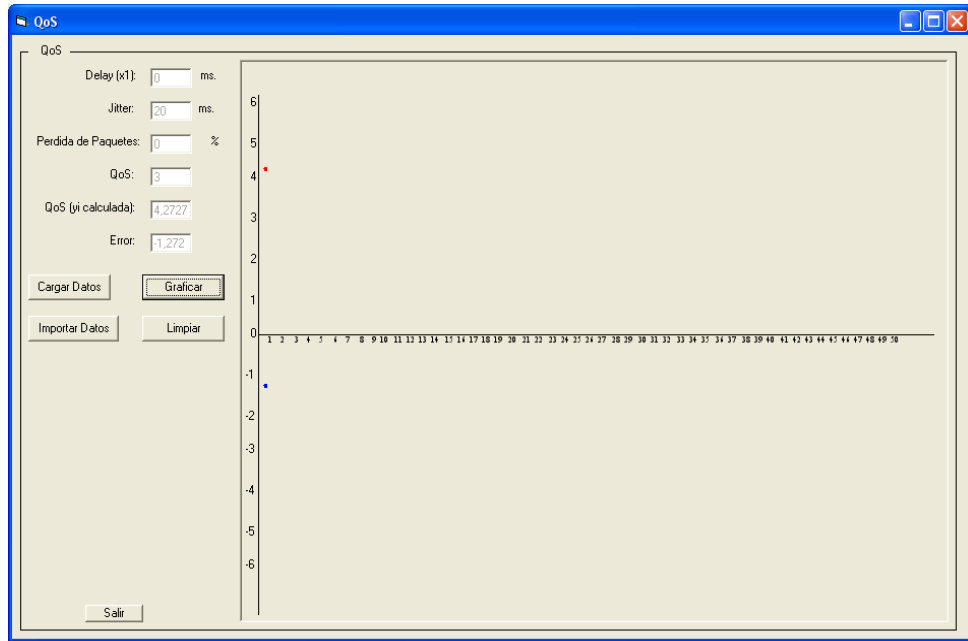


Figura 12. Datos y Gráfica de QoS calculada y el error de una llamada

Al pulsar el botón importar datos se puede realizar la simulación de un conjunto de valores monitoreados, mediante la llamada a un archivo en Excel llamado Error.xls el cual va mostrando cada llamada con la gráfica de la QoS calculada y el error.

Con base a la data recolectada que se muestra en las Tablas 3, 4, 5 6 y 7 se generan las siguientes gráficas:

Lunes 16/07/07

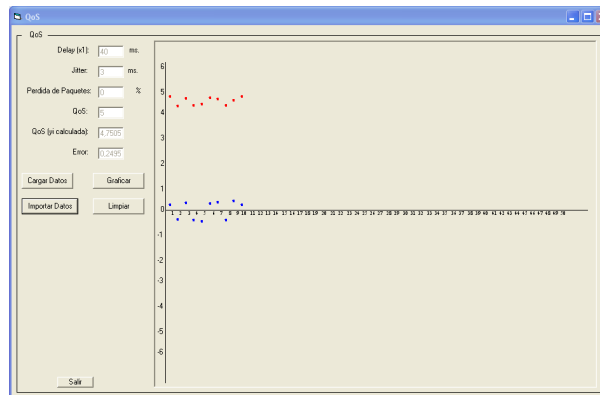


Figura 13. Datos y Gráfica de QoS para los datos del día lunes 16/07/07

Martes 17/07/07

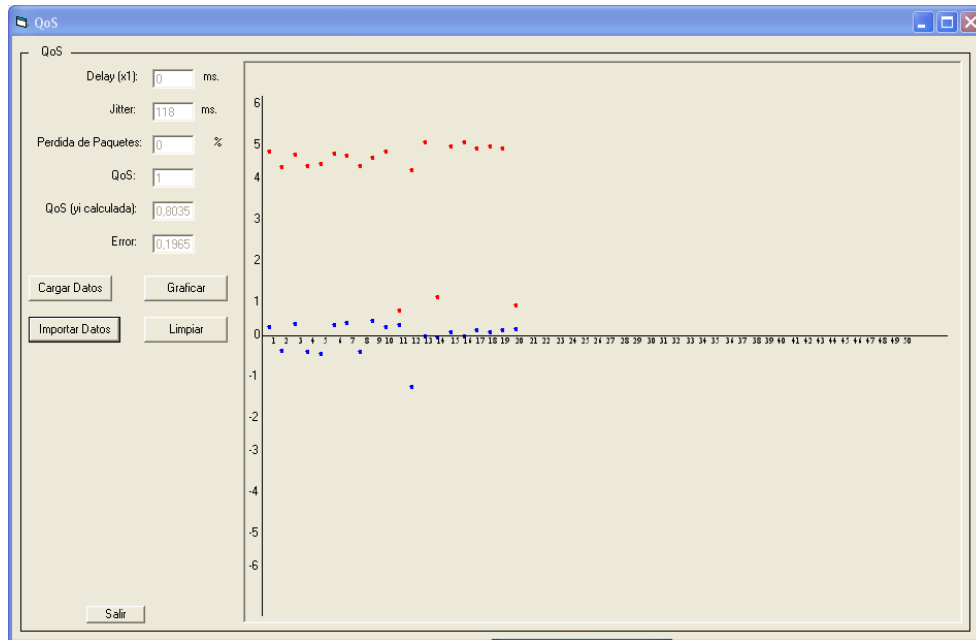


Figura 14. Datos y Gráfica de QoS para los datos del día martes 17/07/07

Miércoles 18/07/07

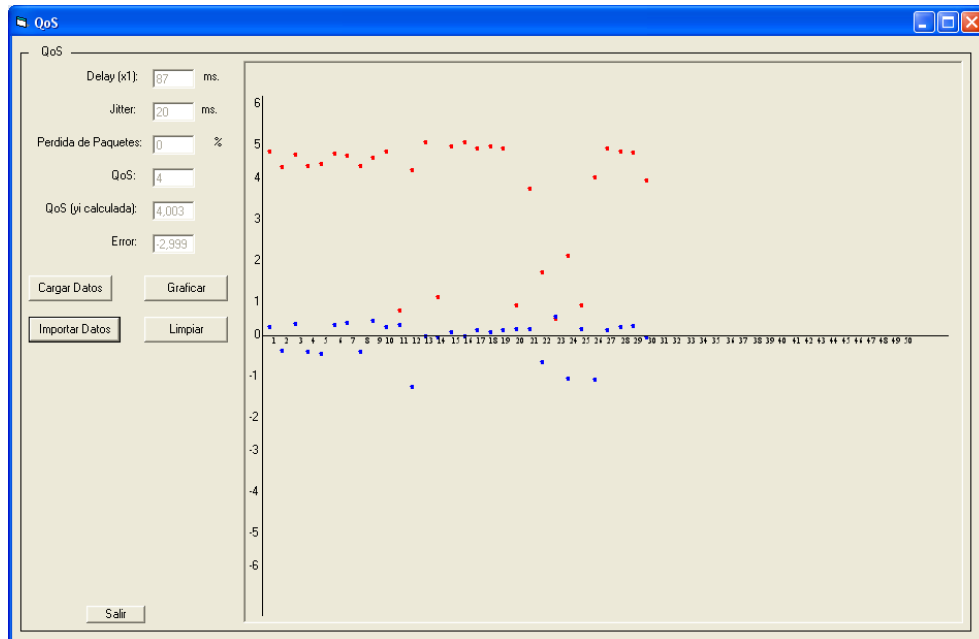


Figura 15. Datos y Gráfica de QoS para los datos del día miércoles 18/07/07

Jueves 19/07/07

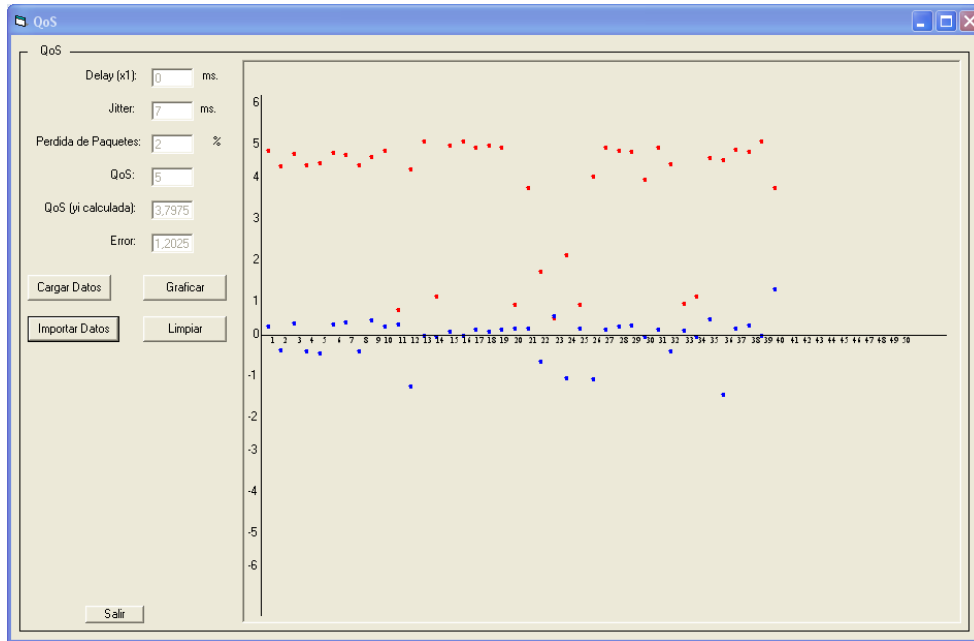


Figura 16. Datos y Gráfica de QoS para los datos del día jueves 19/07/07

Viernes 20/07/07

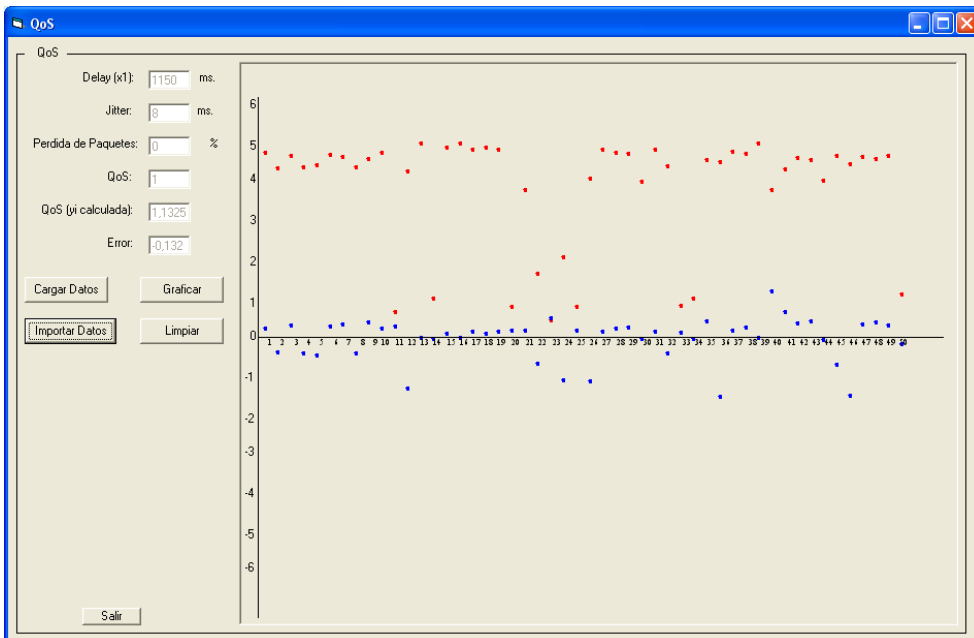


Figura 17. Datos y Gráfica de QoS para los datos del día viernes 20/07/07

Así, cada grupo de valores de llamadas almacenadas en un archivo de Excel puede importarse para analizar la QoS, proceso que debe ser automatizado para leer en tiempo real los datos del equipo de comunicación.

El botón limpiar, deja la pantalla QoS sin datos para realizar un nuevo monitoreo y finalmente el Botón Salir permite abandonar el sistema. El código del programa se puede visualizar en el Anexo E.

Referencias Bibliográficas

- Acquatella Maxime y Alexei Montenegro Robles (2002). Diseño para la implementación de Voz sobre IP en la banda MMDS [Tesis Pregrado]. Universidad Metropolitana. [Consulta: 2005, Agosto 25].
- Ares Roberto (2006). Telefonía-IP. [Consulta en línea]. Disponible: <http://www.monografias.com/trabajos16/telefoniasenalicacion/telefoniasenalicacion.shtml>. [Consulta: 2006, Agosto 15].
- Bastero, Luisa (2005). VoIP en la red del operador. URL: <http://www.aslan.es/boletin/boletin30/acterna.shtml>. (Consulta: Octubre 06, 2005).
- Balestrini Mirian (2001). Como se elabora el proyecto de investigación. BL Consultores Asociados. Caracas
- Barrios Rosa (2004). Estudio y definición de los parámetros e indicadores de calidad de servicio de Internet a ser aplicados en Venezuela [Tesis Pregrado]. Universidad Central de Venezuela. [Consulta 2007, Marzo 29].
- Belzarena Pablo y otros (2006). Metronet: software para medición de calidad de servicio en voz y video [Proyecto]. Universidad de la República, Montevideo, Uruguay.
Disponible: <http://iie.fing.edu.uy/investigacion/grupos/artes/publicaciones/PaperIngSoftV6.pdf>
[Consulta: 2007, Marzo 28].
- Bellido Triana Luis (2004). Contribución a las metodologías para la evaluación de la calidad de servicio en redes heterogeneas [Tesis Doctoral]. Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid. Disponible: <http://greco.dit.upm.es/~lbt/tesis/tesis.pdf>. [Consulta: 2005, Agosto 2005].
- Bravo Montilla Alberto (2005). Calidad de Servicio en Redes de Transmisión de Datos. Manual de Curso dictado por CONATEL.
- Davidson Jonathan y James Peters (2001). Fundamentos de Voz sobre IP. Pearson Educación, S.A., Madrid.
- Descripción técnica detallada sobre Voz sobre IP (VOIP) (2006) [Documento en línea] Disponible: <http://www.monografias.com/trabajos11/descripip/descripip.shtml>

Huidobro José Manuel (2006). La telefonía sobre IP. [Documento en línea].
Disponible: <http://www.monografias.com/trabajos10/tele/tele.shtml>

Hanselman Duane (1996). Matlab edición de estudiante. Prentice Hall. España.

López Jorge y otros (2005). Metodología para la evaluación de servicios de Telecomunicación desde la perspectiva del usuario [Ponencia]. Universidad Politécnica de Madrid.

Disponible: <http://jungla.dit.upm.es/~jlopez/publicaciones/telecomid04lbt.pdf>
[Consulta: 2005, Octubre 15].

Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales (2004). Fondo Editorial de la Universidad Pedagógica Experimental Libertador.

Myers Walpole (2004). Probabilidad y estadística. Cuarta edición. Mc Graw Hill. México.

Pons Alvarez Daniel (2002). Diseño y Desarrollo de un Sistema Experimental Cuantificador de Tráfico de Voz sobre IP (VoIP) [Tesis Pregrado]. Universidad Católica Andrés Bello [Consulta: 2006, Septiembre 2006].

Red Digital de Intercable (2004). Disponible: Departamento de Ingeniería de Intercable.

Recursos VoIP (2005). Introducción VoIP.

Enlace Disponible: <http://www.recursosvoip.com/intro/index.php>

Robins Marc. “Casp (Communications application service provider) Rebirth”, p.35. Internet Telephony. 2005. Volumen 1, número 1. Miami, Florida, USA. Enero 2005.

Rojano Ruiz Elio (2006). Introducción a la VoIP con Linux Asterisk. Segundas Jornadas Técnicas de Linux-Malaga. [Documento en línea] Enlace Disponible: http://jornadas.linux-malaga.org/material1/VoIP_con_Asterisk.pdf#search=%22rojano%20Introducci%C3%B3n%20a%20la%20VoIP%20con%20Linux%20Asterisk%22 [Consulta: 2006, Septiembre 15].

Universidad Centroccidental “Lisandro Alvarado” (UCLA). 2002. Manual para la Elaboración del Trabajo Conducente al Grado Académico de: Especialización, Maestría y Doctorado.

Zaragoza Francisco. “Reduce la factura de teléfono llamando por Internet”. p.4. Telefonía Web. Año 2, Numero 13. España. Editorial Digerati. 2007.

ANEXO A
Lecturas SNMP obtenidas en la Red de Telefonía Digital de Intercable

Lunes.txt

***** SNMP QUERY STARTED *****

```
1: saMtaCallStatLast10EndPoint.1 (integer) 1
2: saMtaCallStatLast10EndPoint.2 (integer) 1
3: saMtaCallStatLast10EndPoint.3 (integer) 1
4: saMtaCallStatLast10EndPoint.4 (integer) 1
5: saMtaCallStatLast10EndPoint.5 (integer) 1
6: saMtaCallStatLast10EndPoint.6 (integer) 1
7: saMtaCallStatLast10EndPoint.7 (integer) 1
8: saMtaCallStatLast10EndPoint.8 (integer) 1
9: saMtaCallStatLast10EndPoint.9 (integer) 1
10: saMtaCallStatLast10EndPoint.10 (integer) 1
11:      saMtaCallStatLast10StartTime.1      (octet      string)
20061002160813Z
12:      saMtaCallStatLast10StartTime.2      (octet      string)
20061002155755Z
13:      saMtaCallStatLast10StartTime.3      (octet      string)
20061002155754Z
14:      saMtaCallStatLast10StartTime.4      (octet      string)
20061002155437Z
15:      saMtaCallStatLast10StartTime.5      (octet      string)
20061002155424Z
16:      saMtaCallStatLast10StartTime.6      (octet      string)
20061002155022Z
17:      saMtaCallStatLast10StartTime.7      (octet      string)
20061002154945Z
18:      saMtaCallStatLast10StartTime.8      (octet      string)
20061002154838Z
19:      saMtaCallStatLast10StartTime.9      (octet      string)
20061002154802Z
20:      saMtaCallStatLast10StartTime.10     (octet      string)
20061002154057Z
21: saMtaCallStatLast10Duration.1 (integer) 8940
22: saMtaCallStatLast10Duration.2 (integer) 5270
23: saMtaCallStatLast10Duration.3 (integer) 1500
24: saMtaCallStatLast10Duration.4 (integer) 5520
```

25: saMtaCallStatLast10Duration.5 (integer) 5680
26: saMtaCallStatLast10Duration.6 (integer) 14360
27: saMtaCallStatLast10Duration.7 (integer) 5600
28: saMtaCallStatLast10Duration.8 (integer) 5690
29: saMtaCallStatLast10Duration.9 (integer) 11540
30: saMtaCallStatLast10Duration.10 (integer) 4880
31: saMtaCallStatLast10Direction.1 (integer) 0
32: saMtaCallStatLast10Direction.2 (integer) 1
33: saMtaCallStatLast10Direction.3 (integer) 1
34: saMtaCallStatLast10Direction.4 (integer) 1
35: saMtaCallStatLast10Direction.5 (integer) 1
36: saMtaCallStatLast10Direction.6 (integer) 0
37: saMtaCallStatLast10Direction.7 (integer) 1
38: saMtaCallStatLast10Direction.8 (integer) 1
39: saMtaCallStatLast10Direction.9 (integer) 1
40: saMtaCallStatLast10Direction.10 (integer) 1
41: saMtaCallStatLast10DialedNumber.1 (octet string)
04164580232 [30.34.31.36.34.35.38.30.32.33.32 (hex)]
42: saMtaCallStatLast10DialedNumber.2 (octet string) (zero-length)
43: saMtaCallStatLast10DialedNumber.3 (octet string)
004164580232 [30.30.34.31.36.34.35.38.30.32.33.32 (hex)]
44: saMtaCallStatLast10DialedNumber.4 (octet string) (zero-length)
45: saMtaCallStatLast10DialedNumber.5 (octet string) 7753
[37.37.35.33 (hex)]
46: saMtaCallStatLast10DialedNumber.6 (octet string) 77982222
[37.37.39.38.32.32.32.32 (hex)]
47: saMtaCallStatLast10DialedNumber.7 (octet string) 773
[37.37.33 (hex)]
48: saMtaCallStatLast10DialedNumber.8 (octet string) 77080898
[37.37.30.38.30.38.39.38 (hex)]
49: saMtaCallStatLast10DialedNumber.9 (octet string) 0 [30 (hex)]
50: saMtaCallStatLast10DialedNumber.10 (octet string) (zero-length)
51: saMtaCallStatLast10UdpPort.1 (integer) 53456
52: saMtaCallStatLast10UdpPort.2 (integer) 53456
53: saMtaCallStatLast10UdpPort.3 (integer) 53456
54: saMtaCallStatLast10UdpPort.4 (integer) 53456
55: saMtaCallStatLast10UdpPort.5 (integer) 53456
56: saMtaCallStatLast10UdpPort.6 (integer) 53456
57: saMtaCallStatLast10UdpPort.7 (integer) 53456

58: saMtaCallStatLast10UdpPort.8 (integer) 53456
59: saMtaCallStatLast10UdpPort.9 (integer) 53456
60: saMtaCallStatLast10UdpPort.10 (integer) 53456
61: saMtaCallStatLast10RemoteIp.1 (octet string) 10.51.226.144
[31.30.2E.35.31.2E.32.32.36.2E.31.34.34 (hex)]
62: saMtaCallStatLast10RemoteIp.2 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
63: saMtaCallStatLast10RemoteIp.3 (octet string) 0.0.0.0
[30.2E.30.2E.30.2E.30 (hex)]
64: saMtaCallStatLast10RemoteIp.4 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
65: saMtaCallStatLast10RemoteIp.5 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
66: saMtaCallStatLast10RemoteIp.6 (octet string) 10.51.231.82
[31.30.2E.35.31.2E.32.33.31.2E.38.32 (hex)]
67: saMtaCallStatLast10RemoteIp.7 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
68: saMtaCallStatLast10RemoteIp.8 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
69: saMtaCallStatLast10RemoteIp.9 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
70: saMtaCallStatLast10RemoteIp.10 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
71: saMtaCallStatLast10RemoteUdpPort.1 (integer) 6690
72: saMtaCallStatLast10RemoteUdpPort.2 (integer) 5940
73: saMtaCallStatLast10RemoteUdpPort.3 (integer) 0
74: saMtaCallStatLast10RemoteUdpPort.4 (integer) 5920
75: saMtaCallStatLast10RemoteUdpPort.5 (integer) 5910
76: saMtaCallStatLast10RemoteUdpPort.6 (integer) 6850
77: saMtaCallStatLast10RemoteUdpPort.7 (integer) 5880
78: saMtaCallStatLast10RemoteUdpPort.8 (integer) 5860
79: saMtaCallStatLast10RemoteUdpPort.9 (integer) 5840
80: saMtaCallStatLast10RemoteUdpPort.10 (integer) 5810
81: saMtaCallStatLast10Packetization.1 (integer) 20
82: saMtaCallStatLast10Packetization.2 (integer) 20
83: saMtaCallStatLast10Packetization.3 (integer) 20
84: saMtaCallStatLast10Packetization.4 (integer) 20
85: saMtaCallStatLast10Packetization.5 (integer) 20
86: saMtaCallStatLast10Packetization.6 (integer) 20
87: saMtaCallStatLast10Packetization.7 (integer) 20

88: saMtaCallStatLast10Packetization.8 (integer) 20
89: saMtaCallStatLast10Packetization.9 (integer) 20
90: saMtaCallStatLast10Packetization.10 (integer) 20
91: saMtaCallStatLast10Codec.1 (integer) pcmu(6)
92: saMtaCallStatLast10Codec.2 (integer) pcmu(6)
93: saMtaCallStatLast10Codec.3 (integer) pcmu(6)
94: saMtaCallStatLast10Codec.4 (integer) pcmu(6)
95: saMtaCallStatLast10Codec.5 (integer) pcmu(6)
96: saMtaCallStatLast10Codec.6 (integer) pcmu(6)
97: saMtaCallStatLast10Codec.7 (integer) pcmu(6)
98: saMtaCallStatLast10Codec.8 (integer) pcmu(6)
99: saMtaCallStatLast10Codec.9 (integer) pcmu(6)
100: saMtaCallStatLast10Codec.10 (integer) pcmu(6)
101: saMtaCallStatLast10SilentSuppression.1 (integer) 0
102: saMtaCallStatLast10SilentSuppression.2 (integer) 0
103: saMtaCallStatLast10SilentSuppression.3 (integer) 0
104: saMtaCallStatLast10SilentSuppression.4 (integer) 0
105: saMtaCallStatLast10SilentSuppression.5 (integer) 0
106: saMtaCallStatLast10SilentSuppression.6 (integer) 0
107: saMtaCallStatLast10SilentSuppression.7 (integer) 0
108: saMtaCallStatLast10SilentSuppression.8 (integer) 0
109: saMtaCallStatLast10SilentSuppression.9 (integer) 0
110: saMtaCallStatLast10SilentSuppression.10 (integer) 0
111: saMtaCallStatLast10EchoCanceller.1 (integer) true(1)
112: saMtaCallStatLast10EchoCanceller.2 (integer) true(1)
113: saMtaCallStatLast10EchoCanceller.3 (integer) true(1)
114: saMtaCallStatLast10EchoCanceller.4 (integer) true(1)
115: saMtaCallStatLast10EchoCanceller.5 (integer) true(1)
116: saMtaCallStatLast10EchoCanceller.6 (integer) true(1)
117: saMtaCallStatLast10EchoCanceller.7 (integer) true(1)
118: saMtaCallStatLast10EchoCanceller.8 (integer) true(1)
119: saMtaCallStatLast10EchoCanceller.9 (integer) true(1)
120: saMtaCallStatLast10EchoCanceller.10 (integer) true(1)
121: saMtaCallStatLast10PacketsSent.1 (integer) 358
122: saMtaCallStatLast10PacketsSent.2 (integer) 24
123: saMtaCallStatLast10PacketsSent.3 (integer) 0
124: saMtaCallStatLast10PacketsSent.4 (integer) 21
125: saMtaCallStatLast10PacketsSent.5 (integer) 0
126: saMtaCallStatLast10PacketsSent.6 (integer) 631
127: saMtaCallStatLast10PacketsSent.7 (integer) 0

128: saMtaCallStatLast10PacketsSent.8 (integer) 0
129: saMtaCallStatLast10PacketsSent.9 (integer) 0
130: saMtaCallStatLast10PacketsSent.10 (integer) 27
131: saMtaCallStatLast10OctetsSent.1 (integer) 57280
132: saMtaCallStatLast10OctetsSent.2 (integer) 3840
133: saMtaCallStatLast10OctetsSent.3 (integer) 0
134: saMtaCallStatLast10OctetsSent.4 (integer) 3360
135: saMtaCallStatLast10OctetsSent.5 (integer) 0
136: saMtaCallStatLast10OctetsSent.6 (integer) 100960
137: saMtaCallStatLast10OctetsSent.7 (integer) 0
138: saMtaCallStatLast10OctetsSent.8 (integer) 0
139: saMtaCallStatLast10OctetsSent.9 (integer) 0
140: saMtaCallStatLast10OctetsSent.10 (integer) 4320
141: saMtaCallStatLast10PacketsReceived.1 (integer) 377
142: saMtaCallStatLast10PacketsReceived.2 (integer) 233
143: saMtaCallStatLast10PacketsReceived.3 (integer) 0
144: saMtaCallStatLast10PacketsReceived.4 (integer) 252
145: saMtaCallStatLast10PacketsReceived.5 (integer) 257
146: saMtaCallStatLast10PacketsReceived.6 (integer) 0
147: saMtaCallStatLast10PacketsReceived.7 (integer) 255
148: saMtaCallStatLast10PacketsReceived.8 (integer) 256
149: saMtaCallStatLast10PacketsReceived.9 (integer) 530
150: saMtaCallStatLast10PacketsReceived.10 (integer) 202
151: saMtaCallStatLast10OctetsReceived.1 (integer) 60320
152: saMtaCallStatLast10OctetsReceived.2 (integer) 37280
153: saMtaCallStatLast10OctetsReceived.3 (integer) 0
154: saMtaCallStatLast10OctetsReceived.4 (integer) 40320
155: saMtaCallStatLast10OctetsReceived.5 (integer) 41120
156: saMtaCallStatLast10OctetsReceived.6 (integer) 0
157: saMtaCallStatLast10OctetsReceived.7 (integer) 40800
158: saMtaCallStatLast10OctetsReceived.8 (integer) 40960
159: saMtaCallStatLast10OctetsReceived.9 (integer) 84800
160: saMtaCallStatLast10OctetsReceived.10 (integer) 32320
161: saMtaCallStatLast10PacketsLost.1 (integer) 0
162: saMtaCallStatLast10PacketsLost.2 (integer) 0
163: saMtaCallStatLast10PacketsLost.3 (integer) 0
164: saMtaCallStatLast10PacketsLost.4 (integer) 0
165: saMtaCallStatLast10PacketsLost.5 (integer) 0
166: saMtaCallStatLast10PacketsLost.6 (integer) 0
167: saMtaCallStatLast10PacketsLost.7 (integer) 0

168: saMtaCallStatLast10PacketsLost.8 (integer) 0
169: saMtaCallStatLast10PacketsLost.9 (integer) 0
170: saMtaCallStatLast10PacketsLost.10 (integer) 0
171: saMtaCallStatLast10Jitter.1 (integer) 3
172: saMtaCallStatLast10Jitter.2 (integer) 11
173: saMtaCallStatLast10Jitter.3 (integer) 5
174: saMtaCallStatLast10Jitter.4 (integer) 10
175: saMtaCallStatLast10Jitter.5 (integer) 8
176: saMtaCallStatLast10Jitter.6 (integer) 4
177: saMtaCallStatLast10Jitter.7 (integer) 6
178: saMtaCallStatLast10Jitter.8 (integer) 9
179: saMtaCallStatLast10Jitter.9 (integer) 7
180: saMtaCallStatLast10Jitter.10 (integer) 3
181: saMtaCallStatLast10Latency.1 (integer) 42
182: saMtaCallStatLast10Latency.2 (integer) 75
183: saMtaCallStatLast10Latency.3 (integer) 45
184: saMtaCallStatLast10Latency.4 (integer) 80
185: saMtaCallStatLast10Latency.5 (integer) 85
186: saMtaCallStatLast10Latency.6 (integer) 48
187: saMtaCallStatLast10Latency.7 (integer) 40
188: saMtaCallStatLast10Latency.8 (integer) 87
189: saMtaCallStatLast10Latency.9 (integer) 46
190: saMtaCallStatLast10Latency.10 (integer) 40
191: saMtaCallStatLast10RemotePacketsSent.1 (integer) 297
192: saMtaCallStatLast10RemotePacketsSent.2 (integer) 1
193: saMtaCallStatLast10RemotePacketsSent.3 (integer) 1
194: saMtaCallStatLast10RemotePacketsSent.4 (integer) 1
195: saMtaCallStatLast10RemotePacketsSent.5 (integer) 1
196: saMtaCallStatLast10RemotePacketsSent.6 (integer) 1
197: saMtaCallStatLast10RemotePacketsSent.7 (integer) 1
198: saMtaCallStatLast10RemotePacketsSent.8 (integer) 138
199: saMtaCallStatLast10RemotePacketsSent.9 (integer) 278
200: saMtaCallStatLast10RemotePacketsSent.10 (integer) 1
201: saMtaCallStatLast10RemoteOctetsSent.1 (integer) 47520
202: saMtaCallStatLast10RemoteOctetsSent.2 (integer) 160
203: saMtaCallStatLast10RemoteOctetsSent.3 (integer) 160
204: saMtaCallStatLast10RemoteOctetsSent.4 (integer) 160
205: saMtaCallStatLast10RemoteOctetsSent.5 (integer) 160
206: saMtaCallStatLast10RemoteOctetsSent.6 (integer) 160
207: saMtaCallStatLast10RemoteOctetsSent.7 (integer) 160

208: saMtaCallStatLast10RemoteOctetsSent.8 (integer) 22080
209: saMtaCallStatLast10RemoteOctetsSent.9 (integer) 44480
210: saMtaCallStatLast10RemoteOctetsSent.10 (integer) 160
211: saMtaCallStatLast10RemotePacketLoss.1 (integer) 6
212: saMtaCallStatLast10RemotePacketLoss.2 (integer) 0
213: saMtaCallStatLast10RemotePacketLoss.3 (integer) 0
214: saMtaCallStatLast10RemotePacketLoss.4 (integer) 0
215: saMtaCallStatLast10RemotePacketLoss.5 (integer) 0
216: saMtaCallStatLast10RemotePacketLoss.6 (integer) 0
217: saMtaCallStatLast10RemotePacketLoss.7 (integer) 0
218: saMtaCallStatLast10RemotePacketLoss.8 (integer) 0
219: saMtaCallStatLast10RemotePacketLoss.9 (integer) 0
220: saMtaCallStatLast10RemotePacketLoss.10 (integer) 0
221: saMtaCallStatLast10RemoteJitter.1 (integer) 0
222: saMtaCallStatLast10RemoteJitter.2 (integer) 0
223: saMtaCallStatLast10RemoteJitter.3 (integer) 0
224: saMtaCallStatLast10RemoteJitter.4 (integer) 0
225: saMtaCallStatLast10RemoteJitter.5 (integer) 0
226: saMtaCallStatLast10RemoteJitter.6 (integer) 0
227: saMtaCallStatLast10RemoteJitter.7 (integer) 0
228: saMtaCallStatLast10RemoteJitter.8 (integer) 0
229: saMtaCallStatLast10RemoteJitter.9 (integer) 0
230: saMtaCallStatLast10RemoteJitter.10 (integer) 0
231: saMtaCallStatLast10BufferUnderruns.1 (integer) 0
232: saMtaCallStatLast10BufferUnderruns.2 (integer) 0
233: saMtaCallStatLast10BufferUnderruns.3 (integer) 0
234: saMtaCallStatLast10BufferUnderruns.4 (integer) 0
235: saMtaCallStatLast10BufferUnderruns.5 (integer) 0
236: saMtaCallStatLast10BufferUnderruns.6 (integer) 0
237: saMtaCallStatLast10BufferUnderruns.7 (integer) 0
238: saMtaCallStatLast10BufferUnderruns.8 (integer) 0
239: saMtaCallStatLast10BufferUnderruns.9 (integer) 0
240: saMtaCallStatLast10BufferUnderruns.10 (integer) 0
241: saMtaCallStatLast10BufferOverruns.1 (integer) 0
242: saMtaCallStatLast10BufferOverruns.2 (integer) 0
243: saMtaCallStatLast10BufferOverruns.3 (integer) 0
244: saMtaCallStatLast10BufferOverruns.4 (integer) 0
245: saMtaCallStatLast10BufferOverruns.5 (integer) 0
246: saMtaCallStatLast10BufferOverruns.6 (integer) 0
247: saMtaCallStatLast10BufferOverruns.7 (integer) 0

248: saMtaCallStatLast10BufferOverruns.8 (integer) 0
249: saMtaCallStatLast10BufferOverruns.9 (integer) 0
250: saMtaCallStatLast10BufferOverruns.10 (integer) 0
251: saMtaCallStatLast10JitterBufferSize.1 (integer) 60
252: saMtaCallStatLast10JitterBufferSize.2 (integer) 60
253: saMtaCallStatLast10JitterBufferSize.3 (integer) 60
254: saMtaCallStatLast10JitterBufferSize.4 (integer) 60
255: saMtaCallStatLast10JitterBufferSize.5 (integer) 60
256: saMtaCallStatLast10JitterBufferSize.6 (integer) 60
257: saMtaCallStatLast10JitterBufferSize.7 (integer) 60
258: saMtaCallStatLast10JitterBufferSize.8 (integer) 60
259: saMtaCallStatLast10JitterBufferSize.9 (integer) 60
260: saMtaCallStatLast10JitterBufferSize.10 (integer) 60
***** SNMP QUERY FINISHED *****

Martes.txt

***** SNMP QUERY STARTED *****
1: saMtaCallStatLast10EndPoint.1 (integer) 1
2: saMtaCallStatLast10EndPoint.2 (integer) 1
3: saMtaCallStatLast10EndPoint.3 (integer) 1
4: saMtaCallStatLast10EndPoint.4 (integer) 1
5: saMtaCallStatLast10EndPoint.5 (integer) 1
6: saMtaCallStatLast10EndPoint.6 (integer) 1
7: saMtaCallStatLast10EndPoint.7 (integer) 1
8: saMtaCallStatLast10EndPoint.8 (integer) 1
9: saMtaCallStatLast10EndPoint.9 (integer) 1
10: saMtaCallStatLast10EndPoint.10 (integer) 1
11: saMtaCallStatLast10StartTime.1 (octet string)
20061002160813Z
12: saMtaCallStatLast10StartTime.2 (octet string)
20061002155755Z
13: saMtaCallStatLast10StartTime.3 (octet string)
20061002155754Z
14: saMtaCallStatLast10StartTime.4 (octet string)
20061002155437Z
15: saMtaCallStatLast10StartTime.5 (octet string)
20061002155424Z
16: saMtaCallStatLast10StartTime.6 (octet string)
20061002155022Z

17: saMtaCallStatLast10StartTime.7 (octet string)
20061002154945Z
18: saMtaCallStatLast10StartTime.8 (octet string)
20061002154838Z
19: saMtaCallStatLast10StartTime.9 (octet string)
20061002154802Z
20: saMtaCallStatLast10StartTime.10 (octet string)
20061002154057Z
21: saMtaCallStatLast10Duration.1 (integer) 8940
22: saMtaCallStatLast10Duration.2 (integer) 5270
23: saMtaCallStatLast10Duration.3 (integer) 1500
24: saMtaCallStatLast10Duration.4 (integer) 5520
25: saMtaCallStatLast10Duration.5 (integer) 5680
26: saMtaCallStatLast10Duration.6 (integer) 14360
27: saMtaCallStatLast10Duration.7 (integer) 5600
28: saMtaCallStatLast10Duration.8 (integer) 5690
29: saMtaCallStatLast10Duration.9 (integer) 11540
30: saMtaCallStatLast10Duration.10 (integer) 4880
31: saMtaCallStatLast10Direction.1 (integer) 0
32: saMtaCallStatLast10Direction.2 (integer) 1
33: saMtaCallStatLast10Direction.3 (integer) 1
34: saMtaCallStatLast10Direction.4 (integer) 1
35: saMtaCallStatLast10Direction.5 (integer) 1
36: saMtaCallStatLast10Direction.6 (integer) 0
37: saMtaCallStatLast10Direction.7 (integer) 1
38: saMtaCallStatLast10Direction.8 (integer) 1
39: saMtaCallStatLast10Direction.9 (integer) 1
40: saMtaCallStatLast10Direction.10 (integer) 1
41: saMtaCallStatLast10DialedNumber.1 (octet string)
04164580232 [30.34.31.36.34.35.38.30.32.33.32 (hex)]
42: saMtaCallStatLast10DialedNumber.2 (octet string) (zero-length)
43: saMtaCallStatLast10DialedNumber.3 (octet string)
004164580232 [30.30.34.31.36.34.35.38.30.32.33.32 (hex)]
44: saMtaCallStatLast10DialedNumber.4 (octet string) (zero-length)
45: saMtaCallStatLast10DialedNumber.5 (octet string) 7753
[37.37.35.33 (hex)]
46: saMtaCallStatLast10DialedNumber.6 (octet string) 77982222
[37.37.39.38.32.32.32.32 (hex)]
47: saMtaCallStatLast10DialedNumber.7 (octet string) 773
[37.37.33 (hex)]

48: saMtaCallStatLast10DialedNumber.8 (octet string) 77080898
[37.37.30.38.30.38.39.38 (hex)]
49: saMtaCallStatLast10DialedNumber.9 (octet string) 0 [30 (hex)]
50: saMtaCallStatLast10DialedNumber.10 (octet string) (zero-length)
51: saMtaCallStatLast10UdpPort.1 (integer) 53456
52: saMtaCallStatLast10UdpPort.2 (integer) 53456
53: saMtaCallStatLast10UdpPort.3 (integer) 53456
54: saMtaCallStatLast10UdpPort.4 (integer) 53456
55: saMtaCallStatLast10UdpPort.5 (integer) 53456
56: saMtaCallStatLast10UdpPort.6 (integer) 53456
57: saMtaCallStatLast10UdpPort.7 (integer) 53456
58: saMtaCallStatLast10UdpPort.8 (integer) 53456
59: saMtaCallStatLast10UdpPort.9 (integer) 53456
60: saMtaCallStatLast10UdpPort.10 (integer) 53456
61: saMtaCallStatLast10RemoteIp.1 (octet string) 10.51.226.144
[31.30.2E.35.31.2E.32.32.36.2E.31.34.34 (hex)]
62: saMtaCallStatLast10RemoteIp.2 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
63: saMtaCallStatLast10RemoteIp.3 (octet string) 0.0.0.0
[30.2E.30.2E.30.2E.30 (hex)]
64: saMtaCallStatLast10RemoteIp.4 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
65: saMtaCallStatLast10RemoteIp.5 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
66: saMtaCallStatLast10RemoteIp.6 (octet string) 10.51.231.82
[31.30.2E.35.31.2E.32.33.31.2E.38.32 (hex)]
67: saMtaCallStatLast10RemoteIp.7 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
68: saMtaCallStatLast10RemoteIp.8 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
69: saMtaCallStatLast10RemoteIp.9 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
70: saMtaCallStatLast10RemoteIp.10 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
71: saMtaCallStatLast10RemoteUdpPort.1 (integer) 6690
72: saMtaCallStatLast10RemoteUdpPort.2 (integer) 5940
73: saMtaCallStatLast10RemoteUdpPort.3 (integer) 0
74: saMtaCallStatLast10RemoteUdpPort.4 (integer) 5920
75: saMtaCallStatLast10RemoteUdpPort.5 (integer) 5910

76: saMtaCallStatLast10RemoteUdpPort.6 (integer) 6850
77: saMtaCallStatLast10RemoteUdpPort.7 (integer) 5880
78: saMtaCallStatLast10RemoteUdpPort.8 (integer) 5860
79: saMtaCallStatLast10RemoteUdpPort.9 (integer) 5840
80: saMtaCallStatLast10RemoteUdpPort.10 (integer) 5810
81: saMtaCallStatLast10Packetization.1 (integer) 20
82: saMtaCallStatLast10Packetization.2 (integer) 20
83: saMtaCallStatLast10Packetization.3 (integer) 20
84: saMtaCallStatLast10Packetization.4 (integer) 20
85: saMtaCallStatLast10Packetization.5 (integer) 20
86: saMtaCallStatLast10Packetization.6 (integer) 20
87: saMtaCallStatLast10Packetization.7 (integer) 20
88: saMtaCallStatLast10Packetization.8 (integer) 20
89: saMtaCallStatLast10Packetization.9 (integer) 20
90: saMtaCallStatLast10Packetization.10 (integer) 20
91: saMtaCallStatLast10Codec.1 (integer) pcmu(6)
92: saMtaCallStatLast10Codec.2 (integer) pcmu(6)
93: saMtaCallStatLast10Codec.3 (integer) pcmu(6)
94: saMtaCallStatLast10Codec.4 (integer) pcmu(6)
95: saMtaCallStatLast10Codec.5 (integer) pcmu(6)
96: saMtaCallStatLast10Codec.6 (integer) pcmu(6)
97: saMtaCallStatLast10Codec.7 (integer) pcmu(6)
98: saMtaCallStatLast10Codec.8 (integer) pcmu(6)
99: saMtaCallStatLast10Codec.9 (integer) pcmu(6)
100: saMtaCallStatLast10Codec.10 (integer) pcmu(6)
101: saMtaCallStatLast10SilentSuppression.1 (integer) 0
102: saMtaCallStatLast10SilentSuppression.2 (integer) 0
103: saMtaCallStatLast10SilentSuppression.3 (integer) 0
104: saMtaCallStatLast10SilentSuppression.4 (integer) 0
105: saMtaCallStatLast10SilentSuppression.5 (integer) 0
106: saMtaCallStatLast10SilentSuppression.6 (integer) 0
107: saMtaCallStatLast10SilentSuppression.7 (integer) 0
108: saMtaCallStatLast10SilentSuppression.8 (integer) 0
109: saMtaCallStatLast10SilentSuppression.9 (integer) 0
110: saMtaCallStatLast10SilentSuppression.10 (integer) 0
111: saMtaCallStatLast10EchoCanceller.1 (integer) true(1)
112: saMtaCallStatLast10EchoCanceller.2 (integer) true(1)
113: saMtaCallStatLast10EchoCanceller.3 (integer) true(1)
114: saMtaCallStatLast10EchoCanceller.4 (integer) true(1)
115: saMtaCallStatLast10EchoCanceller.5 (integer) true(1)

116: saMtaCallStatLast10EchoCanceller.6 (integer) true(1)
117: saMtaCallStatLast10EchoCanceller.7 (integer) true(1)
118: saMtaCallStatLast10EchoCanceller.8 (integer) true(1)
119: saMtaCallStatLast10EchoCanceller.9 (integer) true(1)
120: saMtaCallStatLast10EchoCanceller.10 (integer) true(1)
121: saMtaCallStatLast10PacketsSent.1 (integer) 358
122: saMtaCallStatLast10PacketsSent.2 (integer) 24
123: saMtaCallStatLast10PacketsSent.3 (integer) 0
124: saMtaCallStatLast10PacketsSent.4 (integer) 21
125: saMtaCallStatLast10PacketsSent.5 (integer) 0
126: saMtaCallStatLast10PacketsSent.6 (integer) 631
127: saMtaCallStatLast10PacketsSent.7 (integer) 0
128: saMtaCallStatLast10PacketsSent.8 (integer) 0
129: saMtaCallStatLast10PacketsSent.9 (integer) 0
130: saMtaCallStatLast10PacketsSent.10 (integer) 27
131: saMtaCallStatLast10OctetsSent.1 (integer) 57280
132: saMtaCallStatLast10OctetsSent.2 (integer) 3840
133: saMtaCallStatLast10OctetsSent.3 (integer) 0
134: saMtaCallStatLast10OctetsSent.4 (integer) 3360
135: saMtaCallStatLast10OctetsSent.5 (integer) 0
136: saMtaCallStatLast10OctetsSent.6 (integer) 100960
137: saMtaCallStatLast10OctetsSent.7 (integer) 0
138: saMtaCallStatLast10OctetsSent.8 (integer) 0
139: saMtaCallStatLast10OctetsSent.9 (integer) 0
140: saMtaCallStatLast10OctetsSent.10 (integer) 4320
141: saMtaCallStatLast10PacketsReceived.1 (integer) 377
142: saMtaCallStatLast10PacketsReceived.2 (integer) 233
143: saMtaCallStatLast10PacketsReceived.3 (integer) 0
144: saMtaCallStatLast10PacketsReceived.4 (integer) 252
145: saMtaCallStatLast10PacketsReceived.5 (integer) 257
146: saMtaCallStatLast10PacketsReceived.6 (integer) 0
147: saMtaCallStatLast10PacketsReceived.7 (integer) 255
148: saMtaCallStatLast10PacketsReceived.8 (integer) 256
149: saMtaCallStatLast10PacketsReceived.9 (integer) 530
150: saMtaCallStatLast10PacketsReceived.10 (integer) 202
151: saMtaCallStatLast10OctetsReceived.1 (integer) 60320
152: saMtaCallStatLast10OctetsReceived.2 (integer) 37280
153: saMtaCallStatLast10OctetsReceived.3 (integer) 0
154: saMtaCallStatLast10OctetsReceived.4 (integer) 40320
155: saMtaCallStatLast10OctetsReceived.5 (integer) 41120

156: saMtaCallStatLast10OctetsReceived.6 (integer) 0
157: saMtaCallStatLast10OctetsReceived.7 (integer) 40800
158: saMtaCallStatLast10OctetsReceived.8 (integer) 40960
159: saMtaCallStatLast10OctetsReceived.9 (integer) 84800
160: saMtaCallStatLast10OctetsReceived.10 (integer) 32320
161: saMtaCallStatLast10PacketsLost.1 (integer) 0
162: saMtaCallStatLast10PacketsLost.2 (integer) 0
163: saMtaCallStatLast10PacketsLost.3 (integer) 0
164: saMtaCallStatLast10PacketsLost.4 (integer) 0
165: saMtaCallStatLast10PacketsLost.5 (integer) 0
166: saMtaCallStatLast10PacketsLost.6 (integer) 0
167: saMtaCallStatLast10PacketsLost.7 (integer) 0
168: saMtaCallStatLast10PacketsLost.8 (integer) 0
169: saMtaCallStatLast10PacketsLost.9 (integer) 0
170: saMtaCallStatLast10PacketsLost.10 (integer) 0
171: saMtaCallStatLast10Jitter.1 (integer) 3
172: saMtaCallStatLast10Jitter.2 (integer) 20
173: saMtaCallStatLast10Jitter.3 (integer) 0
174: saMtaCallStatLast10Jitter.4 (integer) 112
175: saMtaCallStatLast10Jitter.5 (integer) 3
176: saMtaCallStatLast10Jitter.6 (integer) 0
177: saMtaCallStatLast10Jitter.7 (integer) 4
178: saMtaCallStatLast10Jitter.8 (integer) 3
179: saMtaCallStatLast10Jitter.9 (integer) 4
180: saMtaCallStatLast10Jitter.10 (integer) 118
181: saMtaCallStatLast10Latency.1 (integer) 1352
182: saMtaCallStatLast10Latency.2 (integer) 0
183: saMtaCallStatLast10Latency.3 (integer) 0
184: saMtaCallStatLast10Latency.4 (integer) 0
185: saMtaCallStatLast10Latency.5 (integer) 0
186: saMtaCallStatLast10Latency.6 (integer) 0
187: saMtaCallStatLast10Latency.7 (integer) 0
188: saMtaCallStatLast10Latency.8 (integer) 0
189: saMtaCallStatLast10Latency.9 (integer) 0
190: saMtaCallStatLast10Latency.10 (integer) 0
191: saMtaCallStatLast10RemotePacketsSent.1 (integer) 297
192: saMtaCallStatLast10RemotePacketsSent.2 (integer) 1
193: saMtaCallStatLast10RemotePacketsSent.3 (integer) 1
194: saMtaCallStatLast10RemotePacketsSent.4 (integer) 1
195: saMtaCallStatLast10RemotePacketsSent.5 (integer) 1

196: saMtaCallStatLast10RemotePacketsSent.6 (integer) 1
197: saMtaCallStatLast10RemotePacketsSent.7 (integer) 1
198: saMtaCallStatLast10RemotePacketsSent.8 (integer) 138
199: saMtaCallStatLast10RemotePacketsSent.9 (integer) 278
200: saMtaCallStatLast10RemotePacketsSent.10 (integer) 1
201: saMtaCallStatLast10RemoteOctetsSent.1 (integer) 47520
202: saMtaCallStatLast10RemoteOctetsSent.2 (integer) 160
203: saMtaCallStatLast10RemoteOctetsSent.3 (integer) 160
204: saMtaCallStatLast10RemoteOctetsSent.4 (integer) 160
205: saMtaCallStatLast10RemoteOctetsSent.5 (integer) 160
206: saMtaCallStatLast10RemoteOctetsSent.6 (integer) 160
207: saMtaCallStatLast10RemoteOctetsSent.7 (integer) 160
208: saMtaCallStatLast10RemoteOctetsSent.8 (integer) 22080
209: saMtaCallStatLast10RemoteOctetsSent.9 (integer) 44480
210: saMtaCallStatLast10RemoteOctetsSent.10 (integer) 160
211: saMtaCallStatLast10RemotePacketLoss.1 (integer) 6
212: saMtaCallStatLast10RemotePacketLoss.2 (integer) 0
213: saMtaCallStatLast10RemotePacketLoss.3 (integer) 0
214: saMtaCallStatLast10RemotePacketLoss.4 (integer) 0
215: saMtaCallStatLast10RemotePacketLoss.5 (integer) 0
216: saMtaCallStatLast10RemotePacketLoss.6 (integer) 0
217: saMtaCallStatLast10RemotePacketLoss.7 (integer) 0
218: saMtaCallStatLast10RemotePacketLoss.8 (integer) 0
219: saMtaCallStatLast10RemotePacketLoss.9 (integer) 0
220: saMtaCallStatLast10RemotePacketLoss.10 (integer) 0
221: saMtaCallStatLast10RemoteJitter.1 (integer) 0
222: saMtaCallStatLast10RemoteJitter.2 (integer) 0
223: saMtaCallStatLast10RemoteJitter.3 (integer) 0
224: saMtaCallStatLast10RemoteJitter.4 (integer) 0
225: saMtaCallStatLast10RemoteJitter.5 (integer) 0
226: saMtaCallStatLast10RemoteJitter.6 (integer) 0
227: saMtaCallStatLast10RemoteJitter.7 (integer) 0
228: saMtaCallStatLast10RemoteJitter.8 (integer) 0
229: saMtaCallStatLast10RemoteJitter.9 (integer) 0
230: saMtaCallStatLast10RemoteJitter.10 (integer) 0
231: saMtaCallStatLast10BufferUnderruns.1 (integer) 0
232: saMtaCallStatLast10BufferUnderruns.2 (integer) 0
233: saMtaCallStatLast10BufferUnderruns.3 (integer) 0
234: saMtaCallStatLast10BufferUnderruns.4 (integer) 0
235: saMtaCallStatLast10BufferUnderruns.5 (integer) 0

236: saMtaCallStatLast10BufferUnderruns.6 (integer) 0
237: saMtaCallStatLast10BufferUnderruns.7 (integer) 0
238: saMtaCallStatLast10BufferUnderruns.8 (integer) 0
239: saMtaCallStatLast10BufferUnderruns.9 (integer) 0
240: saMtaCallStatLast10BufferUnderruns.10 (integer) 0
241: saMtaCallStatLast10BufferOverruns.1 (integer) 0
242: saMtaCallStatLast10BufferOverruns.2 (integer) 0
243: saMtaCallStatLast10BufferOverruns.3 (integer) 0
244: saMtaCallStatLast10BufferOverruns.4 (integer) 0
245: saMtaCallStatLast10BufferOverruns.5 (integer) 0
246: saMtaCallStatLast10BufferOverruns.6 (integer) 0
247: saMtaCallStatLast10BufferOverruns.7 (integer) 0
248: saMtaCallStatLast10BufferOverruns.8 (integer) 0
249: saMtaCallStatLast10BufferOverruns.9 (integer) 0
250: saMtaCallStatLast10BufferOverruns.10 (integer) 0
251: saMtaCallStatLast10JitterBufferSize.1 (integer) 60
252: saMtaCallStatLast10JitterBufferSize.2 (integer) 60
253: saMtaCallStatLast10JitterBufferSize.3 (integer) 60
254: saMtaCallStatLast10JitterBufferSize.4 (integer) 60
255: saMtaCallStatLast10JitterBufferSize.5 (integer) 60
256: saMtaCallStatLast10JitterBufferSize.6 (integer) 60
257: saMtaCallStatLast10JitterBufferSize.7 (integer) 60
258: saMtaCallStatLast10JitterBufferSize.8 (integer) 60
259: saMtaCallStatLast10JitterBufferSize.9 (integer) 60
260: saMtaCallStatLast10JitterBufferSize.10 (integer) 60
***** SNMP QUERY FINISHED *****

Miercoles.txt

***** SNMP QUERY STARTED *****

1: saMtaCallStatLast10EndPoint.1 (integer) 1
2: saMtaCallStatLast10EndPoint.2 (integer) 1
3: saMtaCallStatLast10EndPoint.3 (integer) 1
4: saMtaCallStatLast10EndPoint.4 (integer) 1
5: saMtaCallStatLast10EndPoint.5 (integer) 1
6: saMtaCallStatLast10EndPoint.6 (integer) 1
7: saMtaCallStatLast10EndPoint.7 (integer) 1
8: saMtaCallStatLast10EndPoint.8 (integer) 1
9: saMtaCallStatLast10EndPoint.9 (integer) 1
10: saMtaCallStatLast10EndPoint.10 (integer) 1

11:	saMtaCallStatLast10StartTime.1	(octet	string)
	20061002160813Z		
12:	saMtaCallStatLast10StartTime.2	(octet	string)
	20061002155755Z		
13:	saMtaCallStatLast10StartTime.3	(octet	string)
	20061002155754Z		
14:	saMtaCallStatLast10StartTime.4	(octet	string)
	20061002155437Z		
15:	saMtaCallStatLast10StartTime.5	(octet	string)
	20061002155424Z		
16:	saMtaCallStatLast10StartTime.6	(octet	string)
	20061002155022Z		
17:	saMtaCallStatLast10StartTime.7	(octet	string)
	20061002154945Z		
18:	saMtaCallStatLast10StartTime.8	(octet	string)
	20061002154838Z		
19:	saMtaCallStatLast10StartTime.9	(octet	string)
	20061002154802Z		
20:	saMtaCallStatLast10StartTime.10	(octet	string)
	20061002154057Z		
21:	saMtaCallStatLast10Duration.1	(integer)	8940
22:	saMtaCallStatLast10Duration.2	(integer)	5270
23:	saMtaCallStatLast10Duration.3	(integer)	1500
24:	saMtaCallStatLast10Duration.4	(integer)	5520
25:	saMtaCallStatLast10Duration.5	(integer)	5680
26:	saMtaCallStatLast10Duration.6	(integer)	14360
27:	saMtaCallStatLast10Duration.7	(integer)	5600
28:	saMtaCallStatLast10Duration.8	(integer)	5690
29:	saMtaCallStatLast10Duration.9	(integer)	11540
30:	saMtaCallStatLast10Duration.10	(integer)	4880
31:	saMtaCallStatLast10Direction.1	(integer)	0
32:	saMtaCallStatLast10Direction.2	(integer)	1
33:	saMtaCallStatLast10Direction.3	(integer)	1
34:	saMtaCallStatLast10Direction.4	(integer)	1
35:	saMtaCallStatLast10Direction.5	(integer)	1
36:	saMtaCallStatLast10Direction.6	(integer)	0
37:	saMtaCallStatLast10Direction.7	(integer)	1
38:	saMtaCallStatLast10Direction.8	(integer)	1
39:	saMtaCallStatLast10Direction.9	(integer)	1
40:	saMtaCallStatLast10Direction.10	(integer)	1

41: saMtaCallStatLast10DialedNumber.1 (octet string) 04164580232 [30.34.31.36.34.35.38.30.32.33.32 (hex)]
42: saMtaCallStatLast10DialedNumber.2 (octet string) (zero-length)
43: saMtaCallStatLast10DialedNumber.3 (octet string) 004164580232 [30.30.34.31.36.34.35.38.30.32.33.32 (hex)]
44: saMtaCallStatLast10DialedNumber.4 (octet string) (zero-length)
45: saMtaCallStatLast10DialedNumber.5 (octet string) 7753 [37.37.35.33 (hex)]
46: saMtaCallStatLast10DialedNumber.6 (octet string) 77982222 [37.37.39.38.32.32.32.32 (hex)]
47: saMtaCallStatLast10DialedNumber.7 (octet string) 773 [37.37.33 (hex)]
48: saMtaCallStatLast10DialedNumber.8 (octet string) 77080898 [37.37.30.38.30.38.39.38 (hex)]
49: saMtaCallStatLast10DialedNumber.9 (octet string) 0 [30 (hex)]
50: saMtaCallStatLast10DialedNumber.10 (octet string) (zero-length)
51: saMtaCallStatLast10UdpPort.1 (integer) 53456
52: saMtaCallStatLast10UdpPort.2 (integer) 53456
53: saMtaCallStatLast10UdpPort.3 (integer) 53456
54: saMtaCallStatLast10UdpPort.4 (integer) 53456
55: saMtaCallStatLast10UdpPort.5 (integer) 53456
56: saMtaCallStatLast10UdpPort.6 (integer) 53456
57: saMtaCallStatLast10UdpPort.7 (integer) 53456
58: saMtaCallStatLast10UdpPort.8 (integer) 53456
59: saMtaCallStatLast10UdpPort.9 (integer) 53456
60: saMtaCallStatLast10UdpPort.10 (integer) 53456
61: saMtaCallStatLast10RemoteIp.1 (octet string) 10.51.226.144 [31.30.2E.35.31.2E.32.32.36.2E.31.34.34 (hex)]
62: saMtaCallStatLast10RemoteIp.2 (octet string) 10.51.226.133 [31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
63: saMtaCallStatLast10RemoteIp.3 (octet string) 0.0.0.0 [30.2E.30.2E.30.2E.30 (hex)]
64: saMtaCallStatLast10RemoteIp.4 (octet string) 10.51.226.133 [31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
65: saMtaCallStatLast10RemoteIp.5 (octet string) 10.51.226.133 [31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
66: saMtaCallStatLast10RemoteIp.6 (octet string) 10.51.231.82 [31.30.2E.35.31.2E.32.33.31.2E.38.32 (hex)]

67: saMtaCallStatLast10RemoteIp.7 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
68: saMtaCallStatLast10RemoteIp.8 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
69: saMtaCallStatLast10RemoteIp.9 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
70: saMtaCallStatLast10RemoteIp.10 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
71: saMtaCallStatLast10RemoteUdpPort.1 (integer) 6690
72: saMtaCallStatLast10RemoteUdpPort.2 (integer) 5940
73: saMtaCallStatLast10RemoteUdpPort.3 (integer) 0
74: saMtaCallStatLast10RemoteUdpPort.4 (integer) 5920
75: saMtaCallStatLast10RemoteUdpPort.5 (integer) 5910
76: saMtaCallStatLast10RemoteUdpPort.6 (integer) 6850
77: saMtaCallStatLast10RemoteUdpPort.7 (integer) 5880
78: saMtaCallStatLast10RemoteUdpPort.8 (integer) 5860
79: saMtaCallStatLast10RemoteUdpPort.9 (integer) 5840
80: saMtaCallStatLast10RemoteUdpPort.10 (integer) 5810
81: saMtaCallStatLast10Packetization.1 (integer) 20
82: saMtaCallStatLast10Packetization.2 (integer) 20
83: saMtaCallStatLast10Packetization.3 (integer) 20
84: saMtaCallStatLast10Packetization.4 (integer) 20
85: saMtaCallStatLast10Packetization.5 (integer) 20
86: saMtaCallStatLast10Packetization.6 (integer) 20
87: saMtaCallStatLast10Packetization.7 (integer) 20
88: saMtaCallStatLast10Packetization.8 (integer) 20
89: saMtaCallStatLast10Packetization.9 (integer) 20
90: saMtaCallStatLast10Packetization.10 (integer) 20
91: saMtaCallStatLast10Codec.1 (integer) pcmu(6)
92: saMtaCallStatLast10Codec.2 (integer) pcmu(6)
93: saMtaCallStatLast10Codec.3 (integer) pcmu(6)
94: saMtaCallStatLast10Codec.4 (integer) pcmu(6)
95: saMtaCallStatLast10Codec.5 (integer) pcmu(6)
96: saMtaCallStatLast10Codec.6 (integer) pcmu(6)
97: saMtaCallStatLast10Codec.7 (integer) pcmu(6)
98: saMtaCallStatLast10Codec.8 (integer) pcmu(6)
99: saMtaCallStatLast10Codec.9 (integer) pcmu(6)
100: saMtaCallStatLast10Codec.10 (integer) pcmu(6)
101: saMtaCallStatLast10SilentSuppression.1 (integer) 0
102: saMtaCallStatLast10SilentSuppression.2 (integer) 0

103: saMtaCallStatLast10SilentSuppression.3 (integer) 0
104: saMtaCallStatLast10SilentSuppression.4 (integer) 0
105: saMtaCallStatLast10SilentSuppression.5 (integer) 0
106: saMtaCallStatLast10SilentSuppression.6 (integer) 0
107: saMtaCallStatLast10SilentSuppression.7 (integer) 0
108: saMtaCallStatLast10SilentSuppression.8 (integer) 0
109: saMtaCallStatLast10SilentSuppression.9 (integer) 0
110: saMtaCallStatLast10SilentSuppression.10 (integer) 0
111: saMtaCallStatLast10EchoCanceller.1 (integer) true(1)
112: saMtaCallStatLast10EchoCanceller.2 (integer) true(1)
113: saMtaCallStatLast10EchoCanceller.3 (integer) true(1)
114: saMtaCallStatLast10EchoCanceller.4 (integer) true(1)
115: saMtaCallStatLast10EchoCanceller.5 (integer) true(1)
116: saMtaCallStatLast10EchoCanceller.6 (integer) true(1)
117: saMtaCallStatLast10EchoCanceller.7 (integer) true(1)
118: saMtaCallStatLast10EchoCanceller.8 (integer) true(1)
119: saMtaCallStatLast10EchoCanceller.9 (integer) true(1)
120: saMtaCallStatLast10EchoCanceller.10 (integer) true(1)
121: saMtaCallStatLast10PacketsSent.1 (integer) 358
122: saMtaCallStatLast10PacketsSent.2 (integer) 24
123: saMtaCallStatLast10PacketsSent.3 (integer) 0
124: saMtaCallStatLast10PacketsSent.4 (integer) 21
125: saMtaCallStatLast10PacketsSent.5 (integer) 0
126: saMtaCallStatLast10PacketsSent.6 (integer) 631
127: saMtaCallStatLast10PacketsSent.7 (integer) 0
128: saMtaCallStatLast10PacketsSent.8 (integer) 0
129: saMtaCallStatLast10PacketsSent.9 (integer) 0
130: saMtaCallStatLast10PacketsSent.10 (integer) 27
131: saMtaCallStatLast10OctetsSent.1 (integer) 57280
132: saMtaCallStatLast10OctetsSent.2 (integer) 3840
133: saMtaCallStatLast10OctetsSent.3 (integer) 0
134: saMtaCallStatLast10OctetsSent.4 (integer) 3360
135: saMtaCallStatLast10OctetsSent.5 (integer) 0
136: saMtaCallStatLast10OctetsSent.6 (integer) 100960
137: saMtaCallStatLast10OctetsSent.7 (integer) 0
138: saMtaCallStatLast10OctetsSent.8 (integer) 0
139: saMtaCallStatLast10OctetsSent.9 (integer) 0
140: saMtaCallStatLast10OctetsSent.10 (integer) 4320
141: saMtaCallStatLast10PacketsReceived.1 (integer) 377
142: saMtaCallStatLast10PacketsReceived.2 (integer) 233

143: saMtaCallStatLast10PacketsReceived.3 (integer) 0
144: saMtaCallStatLast10PacketsReceived.4 (integer) 252
145: saMtaCallStatLast10PacketsReceived.5 (integer) 257
146: saMtaCallStatLast10PacketsReceived.6 (integer) 0
147: saMtaCallStatLast10PacketsReceived.7 (integer) 255
148: saMtaCallStatLast10PacketsReceived.8 (integer) 256
149: saMtaCallStatLast10PacketsReceived.9 (integer) 530
150: saMtaCallStatLast10PacketsReceived.10 (integer) 202
151: saMtaCallStatLast10OctetsReceived.1 (integer) 60320
152: saMtaCallStatLast10OctetsReceived.2 (integer) 37280
153: saMtaCallStatLast10OctetsReceived.3 (integer) 0
154: saMtaCallStatLast10OctetsReceived.4 (integer) 40320
155: saMtaCallStatLast10OctetsReceived.5 (integer) 41120
156: saMtaCallStatLast10OctetsReceived.6 (integer) 0
157: saMtaCallStatLast10OctetsReceived.7 (integer) 40800
158: saMtaCallStatLast10OctetsReceived.8 (integer) 40960
159: saMtaCallStatLast10OctetsReceived.9 (integer) 84800
160: saMtaCallStatLast10OctetsReceived.10 (integer) 32320
161: saMtaCallStatLast10PacketsLost.1 (integer) 1
162: saMtaCallStatLast10PacketsLost.2 (integer) 5
163: saMtaCallStatLast10PacketsLost.3 (integer) 0
164: saMtaCallStatLast10PacketsLost.4 (integer) 0
165: saMtaCallStatLast10PacketsLost.5 (integer) 0
166: saMtaCallStatLast10PacketsLost.6 (integer) 0
167: saMtaCallStatLast10PacketsLost.7 (integer) 0
168: saMtaCallStatLast10PacketsLost.8 (integer) 0
169: saMtaCallStatLast10PacketsLost.9 (integer) 0
170: saMtaCallStatLast10PacketsLost.10 (integer) 0
171: saMtaCallStatLast10Jitter.1 (integer) 20
172: saMtaCallStatLast10Jitter.2 (integer) 20
173: saMtaCallStatLast10Jitter.3 (integer) 20
174: saMtaCallStatLast10Jitter.4 (integer) 7
175: saMtaCallStatLast10Jitter.5 (integer) 118
176: saMtaCallStatLast10Jitter.6 (integer) 25
177: saMtaCallStatLast10Jitter.7 (integer) 4
178: saMtaCallStatLast10Jitter.8 (integer) 5
179: saMtaCallStatLast10Jitter.9 (integer) 7
180: saMtaCallStatLast10Jitter.10 (integer) 20
181: saMtaCallStatLast10Latency.1 (integer) 0
182: saMtaCallStatLast10Latency.2 (integer) 90

183: saMtaCallStatLast10Latency.3 (integer) 1225
184: saMtaCallStatLast10Latency.4 (integer) 860
185: saMtaCallStatLast10Latency.5 (integer) 0
186: saMtaCallStatLast10Latency.6 (integer) 0
187: saMtaCallStatLast10Latency.7 (integer) 0
188: saMtaCallStatLast10Latency.8 (integer) 20
189: saMtaCallStatLast10Latency.9 (integer) 0
190: saMtaCallStatLast10Latency.10 (integer) 87
191: saMtaCallStatLast10RemotePacketsSent.1 (integer) 297
192: saMtaCallStatLast10RemotePacketsSent.2 (integer) 1
193: saMtaCallStatLast10RemotePacketsSent.3 (integer) 1
194: saMtaCallStatLast10RemotePacketsSent.4 (integer) 1
195: saMtaCallStatLast10RemotePacketsSent.5 (integer) 1
196: saMtaCallStatLast10RemotePacketsSent.6 (integer) 1
197: saMtaCallStatLast10RemotePacketsSent.7 (integer) 1
198: saMtaCallStatLast10RemotePacketsSent.8 (integer) 138
199: saMtaCallStatLast10RemotePacketsSent.9 (integer) 278
200: saMtaCallStatLast10RemotePacketsSent.10 (integer) 1
201: saMtaCallStatLast10RemoteOctetsSent.1 (integer) 47520
202: saMtaCallStatLast10RemoteOctetsSent.2 (integer) 160
203: saMtaCallStatLast10RemoteOctetsSent.3 (integer) 160
204: saMtaCallStatLast10RemoteOctetsSent.4 (integer) 160
205: saMtaCallStatLast10RemoteOctetsSent.5 (integer) 160
206: saMtaCallStatLast10RemoteOctetsSent.6 (integer) 160
207: saMtaCallStatLast10RemoteOctetsSent.7 (integer) 160
208: saMtaCallStatLast10RemoteOctetsSent.8 (integer) 22080
209: saMtaCallStatLast10RemoteOctetsSent.9 (integer) 44480
210: saMtaCallStatLast10RemoteOctetsSent.10 (integer) 160
211: saMtaCallStatLast10RemotePacketLoss.1 (integer) 6
212: saMtaCallStatLast10RemotePacketLoss.2 (integer) 0
213: saMtaCallStatLast10RemotePacketLoss.3 (integer) 0
214: saMtaCallStatLast10RemotePacketLoss.4 (integer) 0
215: saMtaCallStatLast10RemotePacketLoss.5 (integer) 0
216: saMtaCallStatLast10RemotePacketLoss.6 (integer) 0
217: saMtaCallStatLast10RemotePacketLoss.7 (integer) 0
218: saMtaCallStatLast10RemotePacketLoss.8 (integer) 0
219: saMtaCallStatLast10RemotePacketLoss.9 (integer) 0
220: saMtaCallStatLast10RemotePacketLoss.10 (integer) 0
221: saMtaCallStatLast10RemoteJitter.1 (integer) 0
222: saMtaCallStatLast10RemoteJitter.2 (integer) 0

223: saMtaCallStatLast10RemoteJitter.3 (integer) 0
224: saMtaCallStatLast10RemoteJitter.4 (integer) 0
225: saMtaCallStatLast10RemoteJitter.5 (integer) 0
226: saMtaCallStatLast10RemoteJitter.6 (integer) 0
227: saMtaCallStatLast10RemoteJitter.7 (integer) 0
228: saMtaCallStatLast10RemoteJitter.8 (integer) 0
229: saMtaCallStatLast10RemoteJitter.9 (integer) 0
230: saMtaCallStatLast10RemoteJitter.10 (integer) 0
231: saMtaCallStatLast10BufferUnderruns.1 (integer) 0
232: saMtaCallStatLast10BufferUnderruns.2 (integer) 0
233: saMtaCallStatLast10BufferUnderruns.3 (integer) 0
234: saMtaCallStatLast10BufferUnderruns.4 (integer) 0
235: saMtaCallStatLast10BufferUnderruns.5 (integer) 0
236: saMtaCallStatLast10BufferUnderruns.6 (integer) 0
237: saMtaCallStatLast10BufferUnderruns.7 (integer) 0
238: saMtaCallStatLast10BufferUnderruns.8 (integer) 0
239: saMtaCallStatLast10BufferUnderruns.9 (integer) 0
240: saMtaCallStatLast10BufferUnderruns.10 (integer) 0
241: saMtaCallStatLast10BufferOverruns.1 (integer) 0
242: saMtaCallStatLast10BufferOverruns.2 (integer) 0
243: saMtaCallStatLast10BufferOverruns.3 (integer) 0
244: saMtaCallStatLast10BufferOverruns.4 (integer) 0
245: saMtaCallStatLast10BufferOverruns.5 (integer) 0
246: saMtaCallStatLast10BufferOverruns.6 (integer) 0
247: saMtaCallStatLast10BufferOverruns.7 (integer) 0
248: saMtaCallStatLast10BufferOverruns.8 (integer) 0
249: saMtaCallStatLast10BufferOverruns.9 (integer) 0
250: saMtaCallStatLast10BufferOverruns.10 (integer) 0
251: saMtaCallStatLast10JitterBufferSize.1 (integer) 60
252: saMtaCallStatLast10JitterBufferSize.2 (integer) 60
253: saMtaCallStatLast10JitterBufferSize.3 (integer) 60
254: saMtaCallStatLast10JitterBufferSize.4 (integer) 60
255: saMtaCallStatLast10JitterBufferSize.5 (integer) 60
256: saMtaCallStatLast10JitterBufferSize.6 (integer) 60
257: saMtaCallStatLast10JitterBufferSize.7 (integer) 60
258: saMtaCallStatLast10JitterBufferSize.8 (integer) 60
259: saMtaCallStatLast10JitterBufferSize.9 (integer) 60
260: saMtaCallStatLast10JitterBufferSize.10 (integer) 60
***** SNMP QUERY FINISHED *****

Jueves.txt

***** SNMP QUERY STARTED *****

1: saMtaCallStatLast10EndPoint.1 (integer) 1
2: saMtaCallStatLast10EndPoint.2 (integer) 1
3: saMtaCallStatLast10EndPoint.3 (integer) 1
4: saMtaCallStatLast10EndPoint.4 (integer) 1
5: saMtaCallStatLast10EndPoint.5 (integer) 1
6: saMtaCallStatLast10EndPoint.6 (integer) 1
7: saMtaCallStatLast10EndPoint.7 (integer) 1
8: saMtaCallStatLast10EndPoint.8 (integer) 1
9: saMtaCallStatLast10EndPoint.9 (integer) 1
10: saMtaCallStatLast10EndPoint.10 (integer) 1
11: saMtaCallStatLast10StartTime.1 (octet string)
20061002160813Z
12: saMtaCallStatLast10StartTime.2 (octet string)
20061002155755Z
13: saMtaCallStatLast10StartTime.3 (octet string)
20061002155754Z
14: saMtaCallStatLast10StartTime.4 (octet string)
20061002155437Z
15: saMtaCallStatLast10StartTime.5 (octet string)
20061002155424Z
16: saMtaCallStatLast10StartTime.6 (octet string)
20061002155022Z
17: saMtaCallStatLast10StartTime.7 (octet string)
20061002154945Z
18: saMtaCallStatLast10StartTime.8 (octet string)
20061002154838Z
19: saMtaCallStatLast10StartTime.9 (octet string)
20061002154802Z
20: saMtaCallStatLast10StartTime.10 (octet string)
20061002154057Z
21: saMtaCallStatLast10Duration.1 (integer) 8940
22: saMtaCallStatLast10Duration.2 (integer) 5270
23: saMtaCallStatLast10Duration.3 (integer) 1500
24: saMtaCallStatLast10Duration.4 (integer) 5520
25: saMtaCallStatLast10Duration.5 (integer) 5680
26: saMtaCallStatLast10Duration.6 (integer) 14360
27: saMtaCallStatLast10Duration.7 (integer) 5600
28: saMtaCallStatLast10Duration.8 (integer) 5690

29: saMtaCallStatLast10Duration.9 (integer) 11540
30: saMtaCallStatLast10Duration.10 (integer) 4880
31: saMtaCallStatLast10Direction.1 (integer) 0
32: saMtaCallStatLast10Direction.2 (integer) 1
33: saMtaCallStatLast10Direction.3 (integer) 1
34: saMtaCallStatLast10Direction.4 (integer) 1
35: saMtaCallStatLast10Direction.5 (integer) 1
36: saMtaCallStatLast10Direction.6 (integer) 0
37: saMtaCallStatLast10Direction.7 (integer) 1
38: saMtaCallStatLast10Direction.8 (integer) 1
39: saMtaCallStatLast10Direction.9 (integer) 1
40: saMtaCallStatLast10Direction.10 (integer) 1
41: saMtaCallStatLast10DialedNumber.1 (octet string)
04164580232 [30.34.31.36.34.35.38.30.32.33.32 (hex)]
42: saMtaCallStatLast10DialedNumber.2 (octet string) (zero-length)
43: saMtaCallStatLast10DialedNumber.3 (octet string)
004164580232 [30.30.34.31.36.34.35.38.30.32.33.32 (hex)]
44: saMtaCallStatLast10DialedNumber.4 (octet string) (zero-length)
45: saMtaCallStatLast10DialedNumber.5 (octet string) 7753
[37.37.35.33 (hex)]
46: saMtaCallStatLast10DialedNumber.6 (octet string) 77982222
[37.37.39.38.32.32.32.32 (hex)]
47: saMtaCallStatLast10DialedNumber.7 (octet string) 773
[37.37.33 (hex)]
48: saMtaCallStatLast10DialedNumber.8 (octet string) 77080898
[37.37.30.38.30.38.39.38 (hex)]
49: saMtaCallStatLast10DialedNumber.9 (octet string) 0 [30 (hex)]
50: saMtaCallStatLast10DialedNumber.10 (octet string) (zero-length)
51: saMtaCallStatLast10UdpPort.1 (integer) 53456
52: saMtaCallStatLast10UdpPort.2 (integer) 53456
53: saMtaCallStatLast10UdpPort.3 (integer) 53456
54: saMtaCallStatLast10UdpPort.4 (integer) 53456
55: saMtaCallStatLast10UdpPort.5 (integer) 53456
56: saMtaCallStatLast10UdpPort.6 (integer) 53456
57: saMtaCallStatLast10UdpPort.7 (integer) 53456
58: saMtaCallStatLast10UdpPort.8 (integer) 53456
59: saMtaCallStatLast10UdpPort.9 (integer) 53456
60: saMtaCallStatLast10UdpPort.10 (integer) 53456

61: saMtaCallStatLast10RemoteIp.1 (octet string) 10.51.226.144
[31.30.2E.35.31.2E.32.32.36.2E.31.34.34 (hex)]
62: saMtaCallStatLast10RemoteIp.2 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
63: saMtaCallStatLast10RemoteIp.3 (octet string) 0.0.0.0
[30.2E.30.2E.30.2E.30 (hex)]
64: saMtaCallStatLast10RemoteIp.4 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
65: saMtaCallStatLast10RemoteIp.5 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
66: saMtaCallStatLast10RemoteIp.6 (octet string) 10.51.231.82
[31.30.2E.35.31.2E.32.33.31.2E.38.32 (hex)]
67: saMtaCallStatLast10RemoteIp.7 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
68: saMtaCallStatLast10RemoteIp.8 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
69: saMtaCallStatLast10RemoteIp.9 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
70: saMtaCallStatLast10RemoteIp.10 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
71: saMtaCallStatLast10RemoteUdpPort.1 (integer) 6690
72: saMtaCallStatLast10RemoteUdpPort.2 (integer) 5940
73: saMtaCallStatLast10RemoteUdpPort.3 (integer) 0
74: saMtaCallStatLast10RemoteUdpPort.4 (integer) 5920
75: saMtaCallStatLast10RemoteUdpPort.5 (integer) 5910
76: saMtaCallStatLast10RemoteUdpPort.6 (integer) 6850
77: saMtaCallStatLast10RemoteUdpPort.7 (integer) 5880
78: saMtaCallStatLast10RemoteUdpPort.8 (integer) 5860
79: saMtaCallStatLast10RemoteUdpPort.9 (integer) 5840
80: saMtaCallStatLast10RemoteUdpPort.10 (integer) 5810
81: saMtaCallStatLast10Packetization.1 (integer) 20
82: saMtaCallStatLast10Packetization.2 (integer) 20
83: saMtaCallStatLast10Packetization.3 (integer) 20
84: saMtaCallStatLast10Packetization.4 (integer) 20
85: saMtaCallStatLast10Packetization.5 (integer) 20
86: saMtaCallStatLast10Packetization.6 (integer) 20
87: saMtaCallStatLast10Packetization.7 (integer) 20
88: saMtaCallStatLast10Packetization.8 (integer) 20
89: saMtaCallStatLast10Packetization.9 (integer) 20
90: saMtaCallStatLast10Packetization.10 (integer) 20

91: saMtaCallStatLast10Codec.1 (integer) pcmu(6)
92: saMtaCallStatLast10Codec.2 (integer) pcmu(6)
93: saMtaCallStatLast10Codec.3 (integer) pcmu(6)
94: saMtaCallStatLast10Codec.4 (integer) pcmu(6)
95: saMtaCallStatLast10Codec.5 (integer) pcmu(6)
96: saMtaCallStatLast10Codec.6 (integer) pcmu(6)
97: saMtaCallStatLast10Codec.7 (integer) pcmu(6)
98: saMtaCallStatLast10Codec.8 (integer) pcmu(6)
99: saMtaCallStatLast10Codec.9 (integer) pcmu(6)
100: saMtaCallStatLast10Codec.10 (integer) pcmu(6)
101: saMtaCallStatLast10SilentSuppression.1 (integer) 0
102: saMtaCallStatLast10SilentSuppression.2 (integer) 0
103: saMtaCallStatLast10SilentSuppression.3 (integer) 0
104: saMtaCallStatLast10SilentSuppression.4 (integer) 0
105: saMtaCallStatLast10SilentSuppression.5 (integer) 0
106: saMtaCallStatLast10SilentSuppression.6 (integer) 0
107: saMtaCallStatLast10SilentSuppression.7 (integer) 0
108: saMtaCallStatLast10SilentSuppression.8 (integer) 0
109: saMtaCallStatLast10SilentSuppression.9 (integer) 0
110: saMtaCallStatLast10SilentSuppression.10 (integer) 0
111: saMtaCallStatLast10EchoCanceller.1 (integer) true(1)
112: saMtaCallStatLast10EchoCanceller.2 (integer) true(1)
113: saMtaCallStatLast10EchoCanceller.3 (integer) true(1)
114: saMtaCallStatLast10EchoCanceller.4 (integer) true(1)
115: saMtaCallStatLast10EchoCanceller.5 (integer) true(1)
116: saMtaCallStatLast10EchoCanceller.6 (integer) true(1)
117: saMtaCallStatLast10EchoCanceller.7 (integer) true(1)
118: saMtaCallStatLast10EchoCanceller.8 (integer) true(1)
119: saMtaCallStatLast10EchoCanceller.9 (integer) true(1)
120: saMtaCallStatLast10EchoCanceller.10 (integer) true(1)
121: saMtaCallStatLast10PacketsSent.1 (integer) 358
122: saMtaCallStatLast10PacketsSent.2 (integer) 24
123: saMtaCallStatLast10PacketsSent.3 (integer) 0
124: saMtaCallStatLast10PacketsSent.4 (integer) 21
125: saMtaCallStatLast10PacketsSent.5 (integer) 0
126: saMtaCallStatLast10PacketsSent.6 (integer) 631
127: saMtaCallStatLast10PacketsSent.7 (integer) 0
128: saMtaCallStatLast10PacketsSent.8 (integer) 0
129: saMtaCallStatLast10PacketsSent.9 (integer) 0
130: saMtaCallStatLast10PacketsSent.10 (integer) 27

131: saMtaCallStatLast10OctetsSent.1 (integer) 57280
132: saMtaCallStatLast10OctetsSent.2 (integer) 3840
133: saMtaCallStatLast10OctetsSent.3 (integer) 0
134: saMtaCallStatLast10OctetsSent.4 (integer) 3360
135: saMtaCallStatLast10OctetsSent.5 (integer) 0
136: saMtaCallStatLast10OctetsSent.6 (integer) 100960
137: saMtaCallStatLast10OctetsSent.7 (integer) 0
138: saMtaCallStatLast10OctetsSent.8 (integer) 0
139: saMtaCallStatLast10OctetsSent.9 (integer) 0
140: saMtaCallStatLast10OctetsSent.10 (integer) 4320
141: saMtaCallStatLast10PacketsReceived.1 (integer) 377
142: saMtaCallStatLast10PacketsReceived.2 (integer) 233
143: saMtaCallStatLast10PacketsReceived.3 (integer) 0
144: saMtaCallStatLast10PacketsReceived.4 (integer) 252
145: saMtaCallStatLast10PacketsReceived.5 (integer) 257
146: saMtaCallStatLast10PacketsReceived.6 (integer) 0
147: saMtaCallStatLast10PacketsReceived.7 (integer) 255
148: saMtaCallStatLast10PacketsReceived.8 (integer) 256
149: saMtaCallStatLast10PacketsReceived.9 (integer) 530
150: saMtaCallStatLast10PacketsReceived.10 (integer) 202
151: saMtaCallStatLast10OctetsReceived.1 (integer) 60320
152: saMtaCallStatLast10OctetsReceived.2 (integer) 37280
153: saMtaCallStatLast10OctetsReceived.3 (integer) 0
154: saMtaCallStatLast10OctetsReceived.4 (integer) 40320
155: saMtaCallStatLast10OctetsReceived.5 (integer) 41120
156: saMtaCallStatLast10OctetsReceived.6 (integer) 0
157: saMtaCallStatLast10OctetsReceived.7 (integer) 40800
158: saMtaCallStatLast10OctetsReceived.8 (integer) 40960
159: saMtaCallStatLast10OctetsReceived.9 (integer) 84800
160: saMtaCallStatLast10OctetsReceived.10 (integer) 32320
161: saMtaCallStatLast10PacketsLost.1 (integer) 1
162: saMtaCallStatLast10PacketsLost.2 (integer) 5
163: saMtaCallStatLast10PacketsLost.3 (integer) 0
164: saMtaCallStatLast10PacketsLost.4 (integer) 0
165: saMtaCallStatLast10PacketsLost.5 (integer) 0
166: saMtaCallStatLast10PacketsLost.6 (integer) 0
167: saMtaCallStatLast10PacketsLost.7 (integer) 0
168: saMtaCallStatLast10PacketsLost.8 (integer) 0
169: saMtaCallStatLast10PacketsLost.9 (integer) 0
170: saMtaCallStatLast10PacketsLost.10 (integer) 2

171: saMtaCallStatLast10Jitter.1 (integer) 0
172: saMtaCallStatLast10Jitter.2 (integer) 10
173: saMtaCallStatLast10Jitter.3 (integer) 3
174: saMtaCallStatLast10Jitter.4 (integer) 112
175: saMtaCallStatLast10Jitter.5 (integer) 8
176: saMtaCallStatLast10Jitter.6 (integer) 3
177: saMtaCallStatLast10Jitter.7 (integer) 3
178: saMtaCallStatLast10Jitter.8 (integer) 4
179: saMtaCallStatLast10Jitter.9 (integer) 0
180: saMtaCallStatLast10Jitter.10 (integer) 7
181: saMtaCallStatLast10Latency.1 (integer) 46
182: saMtaCallStatLast10Latency.2 (integer) 75
183: saMtaCallStatLast10Latency.3 (integer) 1300
184: saMtaCallStatLast10Latency.4 (integer) 0
185: saMtaCallStatLast10Latency.5 (integer) 42
186: saMtaCallStatLast10Latency.6 (integer) 120
187: saMtaCallStatLast10Latency.7 (integer) 28
188: saMtaCallStatLast10Latency.8 (integer) 35
189: saMtaCallStatLast10Latency.9 (integer) 0
190: saMtaCallStatLast10Latency.10 (integer) 0
191: saMtaCallStatLast10RemotePacketsSent.1 (integer) 297
192: saMtaCallStatLast10RemotePacketsSent.2 (integer) 1
193: saMtaCallStatLast10RemotePacketsSent.3 (integer) 1
194: saMtaCallStatLast10RemotePacketsSent.4 (integer) 1
195: saMtaCallStatLast10RemotePacketsSent.5 (integer) 1
196: saMtaCallStatLast10RemotePacketsSent.6 (integer) 1
197: saMtaCallStatLast10RemotePacketsSent.7 (integer) 1
198: saMtaCallStatLast10RemotePacketsSent.8 (integer) 138
199: saMtaCallStatLast10RemotePacketsSent.9 (integer) 278
200: saMtaCallStatLast10RemotePacketsSent.10 (integer) 1
201: saMtaCallStatLast10RemoteOctetsSent.1 (integer) 47520
202: saMtaCallStatLast10RemoteOctetsSent.2 (integer) 160
203: saMtaCallStatLast10RemoteOctetsSent.3 (integer) 160
204: saMtaCallStatLast10RemoteOctetsSent.4 (integer) 160
205: saMtaCallStatLast10RemoteOctetsSent.5 (integer) 160
206: saMtaCallStatLast10RemoteOctetsSent.6 (integer) 160
207: saMtaCallStatLast10RemoteOctetsSent.7 (integer) 160
208: saMtaCallStatLast10RemoteOctetsSent.8 (integer) 22080
209: saMtaCallStatLast10RemoteOctetsSent.9 (integer) 44480
210: saMtaCallStatLast10RemoteOctetsSent.10 (integer) 160

211: saMtaCallStatLast10RemotePacketLoss.1 (integer) 6
212: saMtaCallStatLast10RemotePacketLoss.2 (integer) 0
213: saMtaCallStatLast10RemotePacketLoss.3 (integer) 0
214: saMtaCallStatLast10RemotePacketLoss.4 (integer) 0
215: saMtaCallStatLast10RemotePacketLoss.5 (integer) 0
216: saMtaCallStatLast10RemotePacketLoss.6 (integer) 0
217: saMtaCallStatLast10RemotePacketLoss.7 (integer) 0
218: saMtaCallStatLast10RemotePacketLoss.8 (integer) 0
219: saMtaCallStatLast10RemotePacketLoss.9 (integer) 0
220: saMtaCallStatLast10RemotePacketLoss.10 (integer) 0
221: saMtaCallStatLast10RemoteJitter.1 (integer) 0
222: saMtaCallStatLast10RemoteJitter.2 (integer) 0
223: saMtaCallStatLast10RemoteJitter.3 (integer) 0
224: saMtaCallStatLast10RemoteJitter.4 (integer) 0
225: saMtaCallStatLast10RemoteJitter.5 (integer) 0
226: saMtaCallStatLast10RemoteJitter.6 (integer) 0
227: saMtaCallStatLast10RemoteJitter.7 (integer) 0
228: saMtaCallStatLast10RemoteJitter.8 (integer) 0
229: saMtaCallStatLast10RemoteJitter.9 (integer) 0
230: saMtaCallStatLast10RemoteJitter.10 (integer) 0
231: saMtaCallStatLast10BufferUnderruns.1 (integer) 0
232: saMtaCallStatLast10BufferUnderruns.2 (integer) 0
233: saMtaCallStatLast10BufferUnderruns.3 (integer) 0
234: saMtaCallStatLast10BufferUnderruns.4 (integer) 0
235: saMtaCallStatLast10BufferUnderruns.5 (integer) 0
236: saMtaCallStatLast10BufferUnderruns.6 (integer) 0
237: saMtaCallStatLast10BufferUnderruns.7 (integer) 0
238: saMtaCallStatLast10BufferUnderruns.8 (integer) 0
239: saMtaCallStatLast10BufferUnderruns.9 (integer) 0
240: saMtaCallStatLast10BufferUnderruns.10 (integer) 0
241: saMtaCallStatLast10BufferOverruns.1 (integer) 0
242: saMtaCallStatLast10BufferOverruns.2 (integer) 0
243: saMtaCallStatLast10BufferOverruns.3 (integer) 0
244: saMtaCallStatLast10BufferOverruns.4 (integer) 0
245: saMtaCallStatLast10BufferOverruns.5 (integer) 0
246: saMtaCallStatLast10BufferOverruns.6 (integer) 0
247: saMtaCallStatLast10BufferOverruns.7 (integer) 0
248: saMtaCallStatLast10BufferOverruns.8 (integer) 0
249: saMtaCallStatLast10BufferOverruns.9 (integer) 0
250: saMtaCallStatLast10BufferOverruns.10 (integer) 0

251: saMtaCallStatLast10JitterBufferSize.1 (integer) 60
252: saMtaCallStatLast10JitterBufferSize.2 (integer) 60
253: saMtaCallStatLast10JitterBufferSize.3 (integer) 60
254: saMtaCallStatLast10JitterBufferSize.4 (integer) 60
255: saMtaCallStatLast10JitterBufferSize.5 (integer) 60
256: saMtaCallStatLast10JitterBufferSize.6 (integer) 60
257: saMtaCallStatLast10JitterBufferSize.7 (integer) 60
258: saMtaCallStatLast10JitterBufferSize.8 (integer) 60
259: saMtaCallStatLast10JitterBufferSize.9 (integer) 60
260: saMtaCallStatLast10JitterBufferSize.10 (integer) 60
***** SNMP QUERY FINISHED *****

Viernes.txt

***** SNMP QUERY STARTED *****
1: saMtaCallStatLast10EndPoint.1 (integer) 1
2: saMtaCallStatLast10EndPoint.2 (integer) 1
3: saMtaCallStatLast10EndPoint.3 (integer) 1
4: saMtaCallStatLast10EndPoint.4 (integer) 1
5: saMtaCallStatLast10EndPoint.5 (integer) 1
6: saMtaCallStatLast10EndPoint.6 (integer) 1
7: saMtaCallStatLast10EndPoint.7 (integer) 1
8: saMtaCallStatLast10EndPoint.8 (integer) 1
9: saMtaCallStatLast10EndPoint.9 (integer) 1
10: saMtaCallStatLast10EndPoint.10 (integer) 1
11: saMtaCallStatLast10StartTime.1 (octet string)
20061002160813Z
12: saMtaCallStatLast10StartTime.2 (octet string)
20061002155755Z
13: saMtaCallStatLast10StartTime.3 (octet string)
20061002155754Z
14: saMtaCallStatLast10StartTime.4 (octet string)
20061002155437Z
15: saMtaCallStatLast10StartTime.5 (octet string)
20061002155424Z
16: saMtaCallStatLast10StartTime.6 (octet string)
20061002155022Z
17: saMtaCallStatLast10StartTime.7 (octet string)
20061002154945Z
18: saMtaCallStatLast10StartTime.8 (octet string)
20061002154838Z

19: saMtaCallStatLast10StartTime.9 (octet string)
20061002154802Z

20: saMtaCallStatLast10StartTime.10 (octet string)
20061002154057Z

21: saMtaCallStatLast10Duration.1 (integer) 8940

22: saMtaCallStatLast10Duration.2 (integer) 5270

23: saMtaCallStatLast10Duration.3 (integer) 1500

24: saMtaCallStatLast10Duration.4 (integer) 5520

25: saMtaCallStatLast10Duration.5 (integer) 5680

26: saMtaCallStatLast10Duration.6 (integer) 14360

27: saMtaCallStatLast10Duration.7 (integer) 5600

28: saMtaCallStatLast10Duration.8 (integer) 5690

29: saMtaCallStatLast10Duration.9 (integer) 11540

30: saMtaCallStatLast10Duration.10 (integer) 4880

31: saMtaCallStatLast10Direction.1 (integer) 0

32: saMtaCallStatLast10Direction.2 (integer) 1

33: saMtaCallStatLast10Direction.3 (integer) 1

34: saMtaCallStatLast10Direction.4 (integer) 1

35: saMtaCallStatLast10Direction.5 (integer) 1

36: saMtaCallStatLast10Direction.6 (integer) 0

37: saMtaCallStatLast10Direction.7 (integer) 1

38: saMtaCallStatLast10Direction.8 (integer) 1

39: saMtaCallStatLast10Direction.9 (integer) 1

40: saMtaCallStatLast10Direction.10 (integer) 1

41: saMtaCallStatLast10DialedNumber.1 (octet string)
04164580232 [30.34.31.36.34.35.38.30.32.33.32 (hex)]

42: saMtaCallStatLast10DialedNumber.2 (octet string) (zero-length)

43: saMtaCallStatLast10DialedNumber.3 (octet string)
004164580232 [30.30.34.31.36.34.35.38.30.32.33.32 (hex)]

44: saMtaCallStatLast10DialedNumber.4 (octet string) (zero-length)

45: saMtaCallStatLast10DialedNumber.5 (octet string) 7753
[37.37.35.33 (hex)]

46: saMtaCallStatLast10DialedNumber.6 (octet string) 77982222
[37.37.39.38.32.32.32.32 (hex)]

47: saMtaCallStatLast10DialedNumber.7 (octet string) 773
[37.37.33 (hex)]

48: saMtaCallStatLast10DialedNumber.8 (octet string) 77080898
[37.37.30.38.30.38.39.38 (hex)]

49: saMtaCallStatLast10DialedNumber.9 (octet string) 0 [30 (hex)]

50: saMtaCallStatLast10DialedNumber.10 (octet string) (zero-length)
51: saMtaCallStatLast10UdpPort.1 (integer) 53456
52: saMtaCallStatLast10UdpPort.2 (integer) 53456
53: saMtaCallStatLast10UdpPort.3 (integer) 53456
54: saMtaCallStatLast10UdpPort.4 (integer) 53456
55: saMtaCallStatLast10UdpPort.5 (integer) 53456
56: saMtaCallStatLast10UdpPort.6 (integer) 53456
57: saMtaCallStatLast10UdpPort.7 (integer) 53456
58: saMtaCallStatLast10UdpPort.8 (integer) 53456
59: saMtaCallStatLast10UdpPort.9 (integer) 53456
60: saMtaCallStatLast10UdpPort.10 (integer) 53456
61: saMtaCallStatLast10RemoteIp.1 (octet string) 10.51.226.144
[31.30.2E.35.31.2E.32.32.36.2E.31.34.34 (hex)]
62: saMtaCallStatLast10RemoteIp.2 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
63: saMtaCallStatLast10RemoteIp.3 (octet string) 0.0.0.0
[30.2E.30.2E.30.2E.30 (hex)]
64: saMtaCallStatLast10RemoteIp.4 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
65: saMtaCallStatLast10RemoteIp.5 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
66: saMtaCallStatLast10RemoteIp.6 (octet string) 10.51.231.82
[31.30.2E.35.31.2E.32.33.31.2E.38.32 (hex)]
67: saMtaCallStatLast10RemoteIp.7 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
68: saMtaCallStatLast10RemoteIp.8 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
69: saMtaCallStatLast10RemoteIp.9 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
70: saMtaCallStatLast10RemoteIp.10 (octet string) 10.51.226.133
[31.30.2E.35.31.2E.32.32.36.2E.31.33.33 (hex)]
71: saMtaCallStatLast10RemoteUdpPort.1 (integer) 6690
72: saMtaCallStatLast10RemoteUdpPort.2 (integer) 5940
73: saMtaCallStatLast10RemoteUdpPort.3 (integer) 0
74: saMtaCallStatLast10RemoteUdpPort.4 (integer) 5920
75: saMtaCallStatLast10RemoteUdpPort.5 (integer) 5910
76: saMtaCallStatLast10RemoteUdpPort.6 (integer) 6850
77: saMtaCallStatLast10RemoteUdpPort.7 (integer) 5880
78: saMtaCallStatLast10RemoteUdpPort.8 (integer) 5860

79: saMtaCallStatLast10RemoteUdpPort.9 (integer) 5840
80: saMtaCallStatLast10RemoteUdpPort.10 (integer) 5810
81: saMtaCallStatLast10Packetization.1 (integer) 20
82: saMtaCallStatLast10Packetization.2 (integer) 20
83: saMtaCallStatLast10Packetization.3 (integer) 20
84: saMtaCallStatLast10Packetization.4 (integer) 20
85: saMtaCallStatLast10Packetization.5 (integer) 20
86: saMtaCallStatLast10Packetization.6 (integer) 20
87: saMtaCallStatLast10Packetization.7 (integer) 20
88: saMtaCallStatLast10Packetization.8 (integer) 20
89: saMtaCallStatLast10Packetization.9 (integer) 20
90: saMtaCallStatLast10Packetization.10 (integer) 20
91: saMtaCallStatLast10Codec.1 (integer) pcmu(6)
92: saMtaCallStatLast10Codec.2 (integer) pcmu(6)
93: saMtaCallStatLast10Codec.3 (integer) pcmu(6)
94: saMtaCallStatLast10Codec.4 (integer) pcmu(6)
95: saMtaCallStatLast10Codec.5 (integer) pcmu(6)
96: saMtaCallStatLast10Codec.6 (integer) pcmu(6)
97: saMtaCallStatLast10Codec.7 (integer) pcmu(6)
98: saMtaCallStatLast10Codec.8 (integer) pcmu(6)
99: saMtaCallStatLast10Codec.9 (integer) pcmu(6)
100: saMtaCallStatLast10Codec.10 (integer) pcmu(6)
101: saMtaCallStatLast10SilentSuppression.1 (integer) 0
102: saMtaCallStatLast10SilentSuppression.2 (integer) 0
103: saMtaCallStatLast10SilentSuppression.3 (integer) 0
104: saMtaCallStatLast10SilentSuppression.4 (integer) 0
105: saMtaCallStatLast10SilentSuppression.5 (integer) 0
106: saMtaCallStatLast10SilentSuppression.6 (integer) 0
107: saMtaCallStatLast10SilentSuppression.7 (integer) 0
108: saMtaCallStatLast10SilentSuppression.8 (integer) 0
109: saMtaCallStatLast10SilentSuppression.9 (integer) 0
110: saMtaCallStatLast10SilentSuppression.10 (integer) 0
111: saMtaCallStatLast10EchoCanceller.1 (integer) true(1)
112: saMtaCallStatLast10EchoCanceller.2 (integer) true(1)
113: saMtaCallStatLast10EchoCanceller.3 (integer) true(1)
114: saMtaCallStatLast10EchoCanceller.4 (integer) true(1)
115: saMtaCallStatLast10EchoCanceller.5 (integer) true(1)
116: saMtaCallStatLast10EchoCanceller.6 (integer) true(1)
117: saMtaCallStatLast10EchoCanceller.7 (integer) true(1)
118: saMtaCallStatLast10EchoCanceller.8 (integer) true(1)

119: saMtaCallStatLast10EchoCanceller.9 (integer) true(1)
120: saMtaCallStatLast10EchoCanceller.10 (integer) true(1)
121: saMtaCallStatLast10PacketsSent.1 (integer) 358
122: saMtaCallStatLast10PacketsSent.2 (integer) 24
123: saMtaCallStatLast10PacketsSent.3 (integer) 0
124: saMtaCallStatLast10PacketsSent.4 (integer) 21
125: saMtaCallStatLast10PacketsSent.5 (integer) 0
126: saMtaCallStatLast10PacketsSent.6 (integer) 631
127: saMtaCallStatLast10PacketsSent.7 (integer) 0
128: saMtaCallStatLast10PacketsSent.8 (integer) 0
129: saMtaCallStatLast10PacketsSent.9 (integer) 0
130: saMtaCallStatLast10PacketsSent.10 (integer) 27
131: saMtaCallStatLast10OctetsSent.1 (integer) 57280
132: saMtaCallStatLast10OctetsSent.2 (integer) 3840
133: saMtaCallStatLast10OctetsSent.3 (integer) 0
134: saMtaCallStatLast10OctetsSent.4 (integer) 3360
135: saMtaCallStatLast10OctetsSent.5 (integer) 0
136: saMtaCallStatLast10OctetsSent.6 (integer) 100960
137: saMtaCallStatLast10OctetsSent.7 (integer) 0
138: saMtaCallStatLast10OctetsSent.8 (integer) 0
139: saMtaCallStatLast10OctetsSent.9 (integer) 0
140: saMtaCallStatLast10OctetsSent.10 (integer) 4320
141: saMtaCallStatLast10PacketsReceived.1 (integer) 377
142: saMtaCallStatLast10PacketsReceived.2 (integer) 233
143: saMtaCallStatLast10PacketsReceived.3 (integer) 0
144: saMtaCallStatLast10PacketsReceived.4 (integer) 252
145: saMtaCallStatLast10PacketsReceived.5 (integer) 257
146: saMtaCallStatLast10PacketsReceived.6 (integer) 0
147: saMtaCallStatLast10PacketsReceived.7 (integer) 255
148: saMtaCallStatLast10PacketsReceived.8 (integer) 256
149: saMtaCallStatLast10PacketsReceived.9 (integer) 530
150: saMtaCallStatLast10PacketsReceived.10 (integer) 202
151: saMtaCallStatLast10OctetsReceived.1 (integer) 60320
152: saMtaCallStatLast10OctetsReceived.2 (integer) 37280
153: saMtaCallStatLast10OctetsReceived.3 (integer) 0
154: saMtaCallStatLast10OctetsReceived.4 (integer) 40320
155: saMtaCallStatLast10OctetsReceived.5 (integer) 41120
156: saMtaCallStatLast10OctetsReceived.6 (integer) 0
157: saMtaCallStatLast10OctetsReceived.7 (integer) 40800
158: saMtaCallStatLast10OctetsReceived.8 (integer) 40960

159: saMtaCallStatLast10OctetsReceived.9 (integer) 84800
160: saMtaCallStatLast10OctetsReceived.10 (integer) 32320
161: saMtaCallStatLast10PacketsLost.1 (integer) 1
162: saMtaCallStatLast10PacketsLost.2 (integer) 0
163: saMtaCallStatLast10PacketsLost.3 (integer) 0
164: saMtaCallStatLast10PacketsLost.4 (integer) 0
165: saMtaCallStatLast10PacketsLost.5 (integer) 0
166: saMtaCallStatLast10PacketsLost.6 (integer) 0
167: saMtaCallStatLast10PacketsLost.7 (integer) 0
168: saMtaCallStatLast10PacketsLost.8 (integer) 0
169: saMtaCallStatLast10PacketsLost.9 (integer) 0
170: saMtaCallStatLast10PacketsLost.10 (integer) 0
171: saMtaCallStatLast10Jitter.1 (integer) 5
172: saMtaCallStatLast10Jitter.2 (integer) 10
173: saMtaCallStatLast10Jitter.3 (integer) 8
174: saMtaCallStatLast10Jitter.4 (integer) 20
175: saMtaCallStatLast10Jitter.5 (integer) 3
176: saMtaCallStatLast10Jitter.6 (integer) 5
177: saMtaCallStatLast10Jitter.7 (integer) 7
178: saMtaCallStatLast10Jitter.8 (integer) 10
179: saMtaCallStatLast10Jitter.9 (integer) 7
180: saMtaCallStatLast10Jitter.10 (integer) 8
181: saMtaCallStatLast10Latency.1 (integer) 0
182: saMtaCallStatLast10Latency.2 (integer) 0
183: saMtaCallStatLast10Latency.3 (integer) 40
184: saMtaCallStatLast10Latency.4 (integer) 75
185: saMtaCallStatLast10Latency.5 (integer) 63
186: saMtaCallStatLast10Latency.6 (integer) 110
187: saMtaCallStatLast10Latency.7 (integer) 26
188: saMtaCallStatLast10Latency.8 (integer) 15
189: saMtaCallStatLast10Latency.9 (integer) 18
190: saMtaCallStatLast10Latency.10 (integer) 1150
191: saMtaCallStatLast10RemotePacketsSent.1 (integer) 297
192: saMtaCallStatLast10RemotePacketsSent.2 (integer) 1
193: saMtaCallStatLast10RemotePacketsSent.3 (integer) 1
194: saMtaCallStatLast10RemotePacketsSent.4 (integer) 1
195: saMtaCallStatLast10RemotePacketsSent.5 (integer) 1
196: saMtaCallStatLast10RemotePacketsSent.6 (integer) 1
197: saMtaCallStatLast10RemotePacketsSent.7 (integer) 1
198: saMtaCallStatLast10RemotePacketsSent.8 (integer) 138

199: saMtaCallStatLast10RemotePacketsSent.9 (integer) 278
200: saMtaCallStatLast10RemotePacketsSent.10 (integer) 1
201: saMtaCallStatLast10RemoteOctetsSent.1 (integer) 47520
202: saMtaCallStatLast10RemoteOctetsSent.2 (integer) 160
203: saMtaCallStatLast10RemoteOctetsSent.3 (integer) 160
204: saMtaCallStatLast10RemoteOctetsSent.4 (integer) 160
205: saMtaCallStatLast10RemoteOctetsSent.5 (integer) 160
206: saMtaCallStatLast10RemoteOctetsSent.6 (integer) 160
207: saMtaCallStatLast10RemoteOctetsSent.7 (integer) 160
208: saMtaCallStatLast10RemoteOctetsSent.8 (integer) 22080
209: saMtaCallStatLast10RemoteOctetsSent.9 (integer) 44480
210: saMtaCallStatLast10RemoteOctetsSent.10 (integer) 160
211: saMtaCallStatLast10RemotePacketLoss.1 (integer) 6
212: saMtaCallStatLast10RemotePacketLoss.2 (integer) 0
213: saMtaCallStatLast10RemotePacketLoss.3 (integer) 0
214: saMtaCallStatLast10RemotePacketLoss.4 (integer) 0
215: saMtaCallStatLast10RemotePacketLoss.5 (integer) 0
216: saMtaCallStatLast10RemotePacketLoss.6 (integer) 0
217: saMtaCallStatLast10RemotePacketLoss.7 (integer) 0
218: saMtaCallStatLast10RemotePacketLoss.8 (integer) 0
219: saMtaCallStatLast10RemotePacketLoss.9 (integer) 0
220: saMtaCallStatLast10RemotePacketLoss.10 (integer) 0
221: saMtaCallStatLast10RemoteJitter.1 (integer) 0
222: saMtaCallStatLast10RemoteJitter.2 (integer) 0
223: saMtaCallStatLast10RemoteJitter.3 (integer) 0
224: saMtaCallStatLast10RemoteJitter.4 (integer) 0
225: saMtaCallStatLast10RemoteJitter.5 (integer) 0
226: saMtaCallStatLast10RemoteJitter.6 (integer) 0
227: saMtaCallStatLast10RemoteJitter.7 (integer) 0
228: saMtaCallStatLast10RemoteJitter.8 (integer) 0
229: saMtaCallStatLast10RemoteJitter.9 (integer) 0
230: saMtaCallStatLast10RemoteJitter.10 (integer) 0
231: saMtaCallStatLast10BufferUnderruns.1 (integer) 0
232: saMtaCallStatLast10BufferUnderruns.2 (integer) 0
233: saMtaCallStatLast10BufferUnderruns.3 (integer) 0
234: saMtaCallStatLast10BufferUnderruns.4 (integer) 0
235: saMtaCallStatLast10BufferUnderruns.5 (integer) 0
236: saMtaCallStatLast10BufferUnderruns.6 (integer) 0
237: saMtaCallStatLast10BufferUnderruns.7 (integer) 0
238: saMtaCallStatLast10BufferUnderruns.8 (integer) 0

239: saMtaCallStatLast10BufferUnderruns.9 (integer) 0
240: saMtaCallStatLast10BufferUnderruns.10 (integer) 0
241: saMtaCallStatLast10BufferOverruns.1 (integer) 0
242: saMtaCallStatLast10BufferOverruns.2 (integer) 0
243: saMtaCallStatLast10BufferOverruns.3 (integer) 0
244: saMtaCallStatLast10BufferOverruns.4 (integer) 0
245: saMtaCallStatLast10BufferOverruns.5 (integer) 0
246: saMtaCallStatLast10BufferOverruns.6 (integer) 0
247: saMtaCallStatLast10BufferOverruns.7 (integer) 0
248: saMtaCallStatLast10BufferOverruns.8 (integer) 0
249: saMtaCallStatLast10BufferOverruns.9 (integer) 0
250: saMtaCallStatLast10BufferOverruns.10 (integer) 0
251: saMtaCallStatLast10JitterBufferSize.1 (integer) 60
252: saMtaCallStatLast10JitterBufferSize.2 (integer) 60
253: saMtaCallStatLast10JitterBufferSize.3 (integer) 60
254: saMtaCallStatLast10JitterBufferSize.4 (integer) 60
255: saMtaCallStatLast10JitterBufferSize.5 (integer) 60
256: saMtaCallStatLast10JitterBufferSize.6 (integer) 60
257: saMtaCallStatLast10JitterBufferSize.7 (integer) 60
258: saMtaCallStatLast10JitterBufferSize.8 (integer) 60
259: saMtaCallStatLast10JitterBufferSize.9 (integer) 60
260: saMtaCallStatLast10JitterBufferSize.10 (integer) 60
***** SNMP QUERY FINISHED *****

ANEXO B

Análisis estadístico de los datos capturados a través del SPSS

Variables introducidas/eliminadas(b)

Modelo	Variables introducidas	Variables eliminadas	Método
1	PP, JITTER, DELAY(a)	.	Introducir

a Todas las variables solicitadas introducidas

b Variable dependiente: QOS

Resumen del modelo

Modelo	R	R cuadrado	R cuadrado corregida	Error típic. de la estimación
1	,938 ^a	,880	,873	,55878

a. Variables predictoras: (Constante), PP, JITTER, DELAY

ANOVA(b)

Modelo		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Regresión	105,657	3	35,219	112,795	,000(a)
	Residual	14,363	46	,312		
	Total	120,020	49			

a Variables predictoras: (Constante), PP, JITTER, DELAY

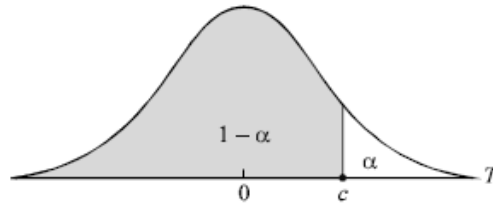
b Variable dependiente: QOS

Coeficientes(a)

Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.
		B	Error típic.	Beta		
1	(Constante)	4,980	,100		49,678	,000
	DELAY	-,003	,000	-,706	-13,722	,000
	JITTER	-,035	,003	-,679	-13,224	,000
	PP	-,468	,103	-,231	-4,525	,000

a Variable dependiente: QOS

ANEXO C
Tabla de distribución t



$1 - \alpha$

n	0.75	0.80	0.85	0.90	0.95	0.975	0.99	0.995
1	1.000	1.376	1.963	3.078	6.314	12.706	31.821	63.657
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921
17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861
20	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845
21	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831
22	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819
23	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807
24	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797
25	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763
29	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756
30	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750
40	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704
60	0.679	0.848	1.046	1.296	1.671	2.000	2.390	2.660
120	0.677	0.845	1.041	1.289	1.658	1.980	2.358	2.617
∞	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576

ANEXO D

Tabla de distribución de F

alfa =	0,05													
	grados de libertad del numerador													
	1	2	3	4	5	6	7	8	9	10	20	60	100	10000
1	161,45	199,5	215,71	224,58	230,16	233,99	236,77	238,88	240,54	241,88	248,02	252,2	253,04	254,3
2	18,513	19	19,164	19,247	19,296	19,329	19,353	19,371	19,385	19,396	19,446	19,479	19,486	19,496
3	10,128	9,5521	9,2766	9,1172	9,0134	8,9407	8,8867	8,8452	8,8123	8,7855	8,6602	8,572	8,5539	8,5267
4	7,7086	6,9443	6,5914	6,3882	6,2561	6,1631	6,0942	6,041	5,9988	5,9644	5,8025	5,6878	5,664	5,6284
5	6,6079	5,7861	5,4094	5,1922	5,0503	4,9503	4,8759	4,8183	4,7725	4,7351	4,5581	4,4314	4,4051	4,3654
6	5,9874	5,1432	4,7571	4,5337	4,3874	4,2839	4,2067	4,1468	4,099	4,06	3,8742	3,7398	3,7117	3,6693
7	5,5915	4,7374	4,3468	4,1203	3,9715	3,866	3,7871	3,7257	3,6767	3,6365	3,4445	3,3043	3,2749	3,2302
8	5,3176	4,459	4,0662	3,8379	3,6875	3,5806	3,5005	3,4381	3,3881	3,3472	3,1503	3,0053	2,9747	2,9281
9	5,1174	4,2565	3,8625	3,6331	3,4817	3,3738	3,2927	3,2296	3,1789	3,1373	2,9365	2,7872	2,7556	2,7072
10	4,9646	4,1028	3,7083	3,478	3,3258	3,2172	3,1355	3,0717	3,0204	2,9782	2,774	2,6211	2,5884	2,5384
11	4,8443	3,9823	3,5874	3,3567	3,2039	3,0946	3,0123	2,948	2,8962	2,8536	2,6464	2,4901	2,4566	2,405
12	4,7472	3,8853	3,4903	3,2592	3,1059	2,9961	2,9134	2,8486	2,7964	2,7534	2,5436	2,3842	2,3498	2,2967
13	4,6672	3,8056	3,4105	3,1791	3,0254	2,9153	2,8321	2,7669	2,7144	2,671	2,4589	2,2966	2,2614	2,207
14	4,6001	3,7389	3,3439	3,1122	2,9582	2,8477	2,7642	2,6987	2,6458	2,6022	2,3879	2,2229	2,187	2,1313
15	4,5431	3,6823	3,2874	3,0556	2,9013	2,7905	2,7066	2,6408	2,5876	2,5437	2,3275	2,1601	2,1234	2,0664
16	4,494	3,6337	3,2389	3,0069	2,8524	2,7413	2,6572	2,5911	2,5377	2,4935	2,2756	2,1058	2,0685	2,0102
17	4,4513	3,5915	3,1968	2,9647	2,81	2,6987	2,6143	2,548	2,4943	2,4499	2,2304	2,0584	2,0204	1,961
18	4,4139	3,5546	3,1599	2,9277	2,7729	2,6613	2,5767	2,5102	2,4563	2,4117	2,1906	2,0166	1,978	1,9175
19	4,3808	3,5219	3,1274	2,8951	2,7401	2,6283	2,5435	2,4768	2,4227	2,3779	2,1555	1,9795	1,9403	1,8787
20	4,3513	3,4928	3,0984	2,8661	2,7109	2,599	2,514	2,4471	2,3928	2,3479	2,1242	1,9464	1,9066	1,8438
21	4,3248	3,4668	3,0725	2,8401	2,6848	2,5727	2,4876	2,4205	2,3661	2,321	2,096	1,9165	1,8761	1,8124
22	4,3009	3,4434	3,0491	2,8167	2,6613	2,5491	2,4638	2,3965	2,3419	2,2967	2,0707	1,8894	1,8486	1,7838
23	4,2793	3,4221	3,028	2,7955	2,64	2,5277	2,4422	2,3748	2,3201	2,2747	2,0476	1,8648	1,8234	1,7577
24	4,2597	3,4028	3,0088	2,7763	2,6207	2,5082	2,4226	2,3551	2,3002	2,2547	2,0267	1,8424	1,8005	1,7338
25	4,2417	3,3852	2,9912	2,7587	2,603	2,4904	2,4047	2,3371	2,2821	2,2365	2,0075	1,8217	1,7794	1,7117
26	4,2252	3,369	2,9752	2,7426	2,5868	2,4741	2,3883	2,3205	2,2655	2,2197	1,9898	1,8027	1,7599	1,6913
27	4,21	3,3541	2,9603	2,7278	2,5719	2,4591	2,3732	2,3053	2,2501	2,2043	1,9736	1,7851	1,7419	1,6724
28	4,196	3,3404	2,9467	2,7141	2,5581	2,4453	2,3593	2,2913	2,236	2,19	1,9586	1,7689	1,7251	1,6548
29	4,183	3,3277	2,934	2,7014	2,5454	2,4324	2,3463	2,2782	2,2229	2,1768	1,9446	1,7537	1,7096	1,6384
30	4,1709	3,3158	2,9223	2,6896	2,5336	2,4205	2,3343	2,2662	2,2107	2,1646	1,9317	1,7396	1,695	1,623
40	4,0847	3,2317	2,8387	2,606	2,4495	2,3359	2,249	2,1802	2,124	2,0773	1,8389	1,6373	1,5892	1,5098
50	4,0343	3,1826	2,79	2,5572	2,4004	2,2864	2,1992	2,1299	2,0733	2,0261	1,7841	1,5757	1,5249	1,4392
60	4,0012	3,1504	2,7581	2,5252	2,3683	2,2541	2,1665	2,097	2,0401	1,9926	1,748	1,5343	1,4814	1,3903
70	3,9778	3,1277	2,7355	2,5027	2,3456	2,2312	2,1435	2,0737	2,0166	1,9689	1,7223	1,5046	1,4498	1,354
80	3,9604	3,1108	2,7188	2,4859	2,3287	2,2142	2,1263	2,0564	1,9991	1,9512	1,7032	1,4821	1,4259	1,3259
90	3,9469	3,0977	2,7058	2,4729	2,3157	2,2011	2,1131	2,043	1,9856	1,9376	1,6883	1,4645	1,407	1,3032
100	3,9362	3,0873	2,6955	2,4626	2,3053	2,1906	2,1025	2,0323	1,9748	1,9267	1,6764	1,4504	1,3917	1,2845
200	3,8884	3,0411	2,6498	2,4168	2,2592	2,1441	2,0556	1,9849	1,9269	1,8783	1,6233	1,3856	1,3206	1,1903
300	3,8726	3,0258	2,6347	2,4017	2,2441	2,1288	2,0402	1,9693	1,9112	1,8623	1,6057	1,3634	1,2958	1,1521
400	3,8648	3,0183	2,6272	2,3943	2,2366	2,1212	2,0325	1,9616	1,9033	1,8544	1,5969	1,3522	1,2831	1,1303
500	3,8601	3,0138	2,6227	2,3898	2,232	2,1167	2,0279	1,9569	1,8986	1,8496	1,5916	1,3455	1,2753	1,1159
600	3,857	3,0107	2,6198	2,3868	2,229	2,1137	2,0248	1,9538	1,8955	1,8465	1,5881	1,341	1,2701	1,1055
700	3,8548	3,0086	2,6176	2,3847	2,2269	2,1115	2,0226	1,9516	1,8932	1,8442	1,5856	1,3377	1,2664	1,0976
800	3,8531	3,007	2,616	2,3831	2,2253	2,1099	2,021	1,95	1,8916	1,8425	1,5837	1,3353	1,2635	1,0912
900	3,8518	3,0057	2,6148	2,3818	2,224	2,1086	2,0197	1,9487	1,8903	1,8412	1,5822	1,3334	1,2613	1,0861

ANEXO E

Software de Simulacion

QoS.form

```
Private Sub Command2_Click()
Dim resp
Dim xl As New Excel.Application
Dim xlw As Excel.Workbook
Set xlw = xl.Workbooks.Open("D:\Nayvic\Tesis\Excel\Error.xls", True, True,, "")
xlw.Sheets("Hoja1").Select
i = 0
resp = 6
While (resp = 6 And i <= 49)
    Delay.Text = xlw.Application.Cells(1 + i, 1).Value
    Jitter.Text = xlw.Application.Cells(1 + i, 2).Value
    PP.Text = xlw.Application.Cells(1 + i, 3).Value
    Text1.Text = xlw.Application.Cells(1 + i, 4).Value
    QoS.Text = xlw.Application.Cells(1 + i, 5).Value
    Text2.Text = xlw.Application.Cells(1 + i, 6).Value
    Q = QoS.Text
    E = Text2.Text
    Picture1.AutoRedraw = True
    x = i / 3
    Picture1.PSet (0.6 + x, 7 - Q), vbRed
    Picture1.PSet (0.6 + x, 7 - E), vbBlue
    resp = MsgBox("¿Monitorear otro dato?", vbQuestion + vbYesNo +
vbDefaultButton1, "")
    i = i + 1
Wend
End Sub

Private Sub Command3_Click()
    CargarDatos.Show
End Sub

Private Sub Calcular_Click()
    Q = QoS.Text
    E = Text2.Text
    Picture1.AutoRedraw = True
    Picture1.PSet (0.6, 7 - Q), vbRed
    Picture1.PSet (0.6, 7 - E), vbBlue
End Sub
```

```
Private Sub Command1_Click()
End
End Sub
```

```
Private Sub Command4_Click()
    Delay.Text = ""
    Jitter.Text = ""
    PP.Text = ""
    Text1.Text = ""
    QoS.Text = ""
    Text2.Text = ""
    Picture1.AutoRedraw = False
    Picture1.BackColor = &H8000000F
End Sub
```

Cargar datos.form

```
Private Sub Command1_Click()
Dim Q, QM As Double
Delay = 0
Jitter = 0
PP = 0
Delay = Text1.Text
Jitter = Text2.Text
PP = Text3.Text
QM = Text4.Text
Q = (4.9807) - (0.0031 * Delay) - (0.0354 * Jitter) - (0.4677 * PP)
QoS.Delay.Text = Text1.Text
QoS.Jitter.Text = Text2.Text
QoS.PP.Text = Text3.Text
QoS.Text1.Text = Text4.Text
QoS.QoS.Text = Q
QoS.Text2.Text = QM - Q
QoS.Show
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        KeyAscii = 0      ' Para que no "pite"
        SendKeys "{tab}"  ' Envía una pulsación TAB
    ElseIf KeyAscii <> 8 Then ' El 8 es la tecla de borrar (backspace)
        ' Si después de añadirle la tecla actual no es un número...
```

```
'If Not IsNumeric("0" & Text1.Text & Chr(KeyAscii)) Then
If Not IsNumeric(Chr(KeyAscii)) Then
    ' ... se desecha esa tecla y se avisa de que no es correcta
    Beep
    KeyAscii = 0
End If
End If
End Sub
```