

**EVALUACIÓN DE UN MODELO DE RED DE COLA CERRADA PARA LA  
ESTIMACIÓN DE LA CAPACIDAD DE CANAL DE UNA RED DE DATOS  
UNIVERSITARIA**

**ELGA XIMENA CARDOZO MORENO**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
FACULTAD DE INGENIERIA ELECTRÓNICA  
BUCARAMANGA**

**2009**

**EVALUACIÓN DE UN MODELO DE RED DE COLA CERRADA PARA LA  
ESTIMACIÓN DE LA CAPACIDAD DE CANAL DE UNA RED DE DATOS  
UNIVERSITARIA**

**ELGA XIMENA CARDOZO MORENO**

**Monografía para optar al título de  
Ingeniero Electrónico**

**Director:**

**JHON JAIRO PADILLA AGUILAR**  
Doctor en Ingeniería Telemática

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
FACULTAD DE INGENIERIA ELECTRÓNICA  
BUCARAMANGA**

**2009**

**A mi madre Elga Lucía Moreno**

## **AGRADECIMIENTOS**

El autor expresa su agradecimiento a:

Jhon Jairo Padilla Aguilar, Doctor en Ingeniería telemática y Director de la Investigación, por sus ingeniosas orientaciones, su constante motivación, su inagotable paciencia y su invaluable confianza en mí, fundamentales para la obtención de los resultados de este trabajo.

Jaime Alberto Moreno, por sus valiosas sugerencias y aportes durante el inicio del desarrollo de este trabajo.

Elga Lucía Moreno, Amanda López y Cecilia Cardozo, por sus oraciones y por compartir los éxitos y fracasos durante la realización del presente trabajo.

Juan Manuel Rojas Forero, por su paciencia y por acompañarme en todos los momentos importantes.

## CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>13</b>
1.1	PLANTEAMIENTO	13
1.2	JUSTIFICACIÓN	14
1.3	OBJETIVOS	15
1.3.1	<i>Objetivo General</i>	15
1.3.2	<i>Objetivos Específicos</i>	15
1.4	METODOLOGIA UTILIZADA	16
<b>2</b>	<b>ESTADO DEL ARTE</b>	<b>22</b>
2.1	MODELADO Y SIMULACIÓN DE SISTEMAS	22
2.1.1	<i>El proceso de modelado</i>	23
2.2	PROCESOS ESTOCÁSTICOS	28
2.2.1	<i>Espacio Parametral</i>	31
2.2.2	<i>Espacio de Estados</i>	31
2.2.3	<i>Ejemplos de procesos aleatorios</i>	34
2.3	TEORÍA DE LÍNEAS DE ESPERA	35
2.3.1	<i>Estructura de una línea de espera</i>	36
2.3.2	<i>Patrón de llegadas</i>	39
2.3.3	<i>Demanda y mecanismo de servicio</i>	42
2.3.4	<i>Disciplina de servicio</i>	44
2.3.5	<i>Definiciones y Notación</i>	46
2.3.6	<i>Fórmula de Little</i>	49
2.3.7	<i>Medidas orientadas al gestor del sistema</i>	50
2.3.8	<i>Medidas orientadas al usuario</i>	51
2.4	DESCRIPCIÓN DEL PROTOCOLO TCP	52
2.4.1	<i>Objetivos de TCP</i>	52
2.4.2	<i>Cabecera TCP</i>	52
2.4.3	<i>Control de Congestión en TCP</i>	57
2.4.4	<i>Trasferencia de archivos con TCP</i>	59
2.5	DESCRIPCIÓN DEL PROTOCOLO HTTP	60
2.5.1	<i>Peticiones HTTP</i>	61
2.5.2	<i>Peticiones HTTP sobre Conexiones TCP</i>	63
2.6	MODELO DE RED DE COLA CERRADA CQN	66
2.6.1	<i>Introducción</i>	66
2.6.2	<i>Descripción del fenómeno a modelar</i>	68
2.6.3	<i>Formalización del modelo de Red de Cola Cerrada</i>	70
2.6.4	<i>Parámetros del Modelo de Red de Cola Cerrada</i>	73
2.6.5	<i>Ancho de Banda Efectivo</i>	75
2.6.6	<i>El criterio del valor medio</i>	76
2.6.7	<i>Solución aproximada por el desarrollo del criterio de media</i>	82
2.7	HERRAMIENTAS DE SOFTWARE UTILIZADAS	84
2.7.1	<i>Wireshark</i>	84
2.7.2	<i>Honeysnap</i>	90

2.7.3	<i>Sed</i> .....	93
2.7.4	<i>Network Simulator 2 (NS-2)</i> .....	95
<b>3</b>	<b>DESARROLLO DE LA TESIS</b> .....	<b>98</b>
3.1	DEFINICIÓN DEL PROBLEMA .....	98
3.1.1	<i>Importancia</i> .....	99
3.1.2	<i>Unidad de Ancho de Banda</i> .....	100
3.1.3	<i>Limitaciones</i> .....	101
3.1.4	<i>Tasa de Transferencia</i> .....	102
3.2	DESCRIPCIÓN DE LA RED DE DATOS UNIVERSITARIA .....	105
3.2.1	<i>Indagar quien tiene la información</i> .....	105
3.2.2	<i>Documentar la información según cada elemento de la red</i> .....	106
3.2.3	<i>Diagrama Lógico de la red de datos UPB seccional Bucaramanga</i> .....	109
3.2.4	<i>Restricciones administrativas sobre la red</i> .....	112
3.3	CARACTERIZACIÓN DE LAS TRAZAS DE TRÁFICO .....	113
3.3.1	<i>Medición del desempeño de la red</i> .....	114
3.3.2	<i>Selección de la muestra</i> .....	115
3.3.3	<i>Obtención de trazas TCP con Wireshark</i> .....	117
3.3.4	<i>Extracción de Conexiones TCP y UDP con Honeysnap</i> .....	117
3.3.5	<i>Organización de la información con Sed</i> .....	120
3.3.6	<i>Resultados de la utilización del canal de Internet con Excel</i> .....	122
3.3.7	<i>Análisis de los resultados obtenidos de la utilización del canal de Internet.</i> .....	129
3.3.8	<i>Comportamiento del tráfico Web</i> .....	130
3.4	SIMULACIÓN DEL COMPORTAMIENTO DE LA RED .....	134
3.4.1	<i>Generalidades</i> .....	134
3.4.2	<i>Tipos de Modelos de Simulación</i> .....	135
3.4.3	<i>Proceso de desarrollo de un modelo de simulación</i> .....	137
3.4.4	<i>Descripción del Modelo de Red en Ns-2</i> .....	141
3.4.5	<i>Resultados obtenidos de la simulación</i> .....	148
3.5	APLICACIÓN DEL MODELO DE RED DE COLA CERRADA .....	150
3.5.1	<i>Construcción del algoritmo en Excel</i> .....	152
3.5.2	<i>Resultados obtenidos para la hora Pico</i> .....	155
3.5.3	<i>Validación del Modelo</i> .....	157
3.6	SINTESIS DE LA METODOLOGÍA USADA .....	158
	<b>CONCLUSIONES</b> .....	<b>160</b>
	<b>BIBLIOGRAFÍA</b> .....	<b>165</b>
	<b>ANEXOS</b> .....	<b>168</b>
A1.	MACRO PARA OBTENER ARCHIVO DE TEXTO Y CALCULAR LOS DIFERENTES PARÁMETROS. ....	168
A2.	MACRO PARA CALCULAR EL ÍNDICE DE PARETO Y PARÁMETRO DE HURST .....	171
A3.	MACRO PARA CALCULAR LA MEDIA DEL TAMAÑO DE ARCHIVO Y LA MEDIA DEL TIEMPO ENTRE CONEXIONES.....	178

## LISTA DE FIGURAS

FIGURA 1: FORMALIZACIÓN DEL PROCESO. ....	23
FIGURA 2: APLICACIÓN DEL DESARROLLO DEL MODELO DE EVALUACIÓN DE UN PROCESO.....	26
FIGURA 3: NÚMERO DE MENSAJES, $X(t)$ , QUE HAY EN EL NODO EN EL INSTANTE $t$ . ....	29
FIGURA 4: ESTACIÓN DE SERVICIO CON $M$ SERVIDORES. ....	36
FIGURA 5: LLEGADA DE TAREAS AL SISTEMA.....	40
FIGURA 6 CABECERA TCP.....	53
FIGURA 7: ESTABLECER UNA CONEXIÓN MEDIANTE EL PROTOCOLO DE ACUERDO DE TRES VÍAS. ....	55
FIGURA 8: LIBERAR UNA CONEXIÓN MEDIANTE EL PROTOCOLO DE ACUERDO DE TRES VÍAS. ....	56
FIGURA 9: ALGORITMO DE CONGESTIONAMIENTO EN INTERNET ....	58
FIGURA 10: FASES DEL ALGORITMO DE ARRANQUE LENTO (SLOW START).....	60
FIGURA 11: PÁGINA HTML CON TRES IMÁGENES. ....	61
FIGURA 12: EJEMPLO DE PETICIÓN HTTP MÉTODO GET ....	62
FIGURA 13: EJEMPLO DE PETICIÓN HTTP MÉTODO PUT ....	63
FIGURA 14: HTTP SOBRE TCP.....	64
FIGURA 15: (A) HTTP 1.0 (B) HTTP 1.1 ....	65
FIGURA 16: ESTRUCTURA DE UNA TRANSACCIÓN WEB.....	70
FIGURA 17: MODELO M/G/N.....	72
FIGURA 18: MODELO M/M/1.....	72
FIGURA 19: MODELO CQN. ....	73
FIGURA 20: EJEMPLO DE LA FORMULA B DE ERLANG. ....	81
FIGURA 21: GUI WIRESHARK. ....	86
FIGURA 22: TOPOLOGÍA PARA LA CAPTURA DE TRÁFICO. ....	87
FIGURA 23: MENÚ CAPTURE.....	88
FIGURA 24: MENÚ CAPTURE INTERFACES.....	88
FIGURA 25: MENÚ CAPTURE OPTIONS. ....	89
FIGURA 26: INGRESO A PYTHON EN LA LÍNEA DE COMANDOS. ....	91
FIGURA 27: ARCHIVOS GENERADOS POR HONEYSNAP. ....	92
FIGURA 28: RESUMEN DE CONEXIONES TCP ARCHIVO INCOMING.TXT. ....	93
FIGURA 29: RESUMEN DE CONEXIONES UDP ARCHIVO OUTGING.TXT. ....	93
FIGURA 30: ARCHIVO PATRON_3.SED . ....	95
FIGURA 31: RED DE DATOS REPRESENTADA COMO UNA AUTOPISTA DE INFORMACIÓN.....	98
FIGURA 32: DIAGRAMA LÓGICO DE LA RED UPB SECCIONAL BUCARAMANGA CANAL TELECOM.....	110
FIGURA 33: SERVIDOR PROXY CON CACHE. ....	111
FIGURA 34: PASOS PARA LA CARACTERIZACIÓN DE LAS TRAZAS DE TRÁFICO. ....	113
FIGURA 35: MRTG CANAL TELECOM TERCERA SEMANA DE MAYO DE 2006. ....	115
FIGURA 36: CONEXIÓN TCP EN WIRESHARK. ....	117
FIGURA 37: EJECUTANDO HONEYSNAP EN LA LÍNEA DE COMANDOS.....	118
FIGURA 38: RESUMEN DE CONEXIONES TCP ARCHIVO INCOMING.TXT. ....	118
FIGURA 39: RESUMEN DE CONEXIONES UDP ARCHIVO OUTGOING.TXT. ....	119
FIGURA 40: ARCHIVOS GENERADOS POR HONEYSNAP. ....	119
FIGURA 41: CONEXIÓN TCP CON HONEYSNAP.....	120
FIGURA 42: LÍNEA DE COMANDOS LDCL.BAT PARA EDITAR INCOMING.TXT. ....	121
FIGURA 43: ARCHIVO PATRON_2.SED.....	121
FIGURA 44: <i>RESULTADOSI.TXT</i> OBTENIDOS CON SED.....	122
FIGURA 45: <i>RESULTADOSO.TXT</i> OBTENIDOS CON SED.....	122
FIGURA 46: TOTAL DE BYTES DE LA RED UPB AL PROXY PANDA EL 18 DE MAYO DE 2006. ....	124

FIGURA 47: TOTAL DE BYTES DE PROXY PANDA A LA RED UPB EL 18 DE MAYO DE 2006.....	125
FIGURA 48: TOTAL DE BYTES DE LA RED UPB AL PROXY PANDA EL 19 DE MAYO DE 2006. ....	126
FIGURA 49: TOTAL DE BYTES DE PROXY PANDA A LA RED UPB EL 19 DE MAYO DE 2006.....	127
FIGURA 50: TOTAL DE BYTES DE LA RED UPB AL PROXY PANDA EL 20 DE MAYO DE 2006. ....	128
FIGURA 51: TOTAL DE BYTES DE PROXY PANDA A LA RED UPB EL 20 DE MAYO DE 2006.....	129
FIGURA 52: DIAGRAMA DE PARETO DEL COMPORTAMIENTO DEL TAMAÑO DE LOS ARCHIVOS.....	132
FIGURA 53: PERDIDA DE PAQUETES EN EL ENLACE QUE REPRESENTA EL CUELLO DE BOTELLA. ....	148
FIGURA 54: NÚMERO DE PAQUETES EN EL ENLACE QUE REPRESENTA EL CUELLO DE BOTELLA. ....	149
FIGURA 55: ANCHO DE BANDA USADO SOBRE EL ENLACE REPRESENTA EL CUELLO DE BOTELLA. ....	150
FIGURA 56: CAPACIDAD NECESARIA EN EL ENLACE DOWNLINK DE TELECOM RESPECTO A LA CACIDAD DE USUARIO. ....	156
FIGURA 57: CAPACIDAD NECESARIA EN EL ENLACE UPLINK DE TELECOM RESPECTO A LA CAPACIDAD DE USUARIO. ....	156
FIGURA 58: ANCHO DE BANDA USADO SOBRE EL ENLACE REPRESENTA EL CUELLO DE BOTELLA Y CAPACIDAD NECESARIA EN EL ENLACE UPLINK DE TELECOM RESPECTO A LA CAPACIDAD DE USUARIO.....	157
FIGURA 59: DIAGRAMA DE FLUJO “METODOLOGÍA PARA LA IMPLEMENTACIÓN DEL MODELO CQN”.....	158
FIGURA 60: DIAGRAMA DE FLUJO “METODOLOGÍA PARA LA IMPLEMENTACIÓN DEL MODELO CQN” PROCESOS 1 Y 2. ....	159



## LISTA DE TABLAS

TABLA 1: EJEMPLOS NOTACIÓN KENDALL.....	48
TABLA 2: APLICACIÓN DEL MVA SOBRE EL MODELO DE RED DE COLA CERRADA DONDE $S=1$ . ....	79
TABLA 3: APLICACIÓN DEL MVA SOBRE EL MODELO DE RED DE COLA CERRADA DONDE $S=2$ . ....	80
TABLA 4: APLICACIÓN DEL MVA SOBRE EL MODELO DE RED DE COLA CERRADA DONDE $S=X$ . ....	80
TABLA 5 UNIDADES DE ANCHO DE BANDA. ....	101
TABLA 6: LIMITACIONES DE LONGITUD Y ANCHOS DE BANDA MÁXIMOS.....	102
TABLA 7: SERVICIOS WAN COMUNES Y ANCHOS DE BANDA ASOCIADOS.....	102
TABLA 8: PERSONAL ENCARGADO DE LA RED DE LA UPB SECCIONAL BUCARAMANGA EN MAYO 2006.....	105
TABLA 9: PERSONAL ENCARGADO DE LA RED DE LA UPB SECCIONAL BUCARAMANGA EN ENERO 2008.....	105
TABLA 10: SERVIDORES DE LA RED DE LA UPB SECCIONAL BUCARAMANGA EN MAYO DE 2006. ....	106
TABLA 11 : MUESTRA INFORMACIÓN DE DISPOSITIVOS. ....	106
TABLA 12: VLANS DE LA RED DE LA UPB SECCIONAL BUCARAMANGA EN MAYO DE 2006. ....	107
TABLA 13: VLANS DE LA RED DE LA UPB SECCIONAL BUCARAMANGA EN ENERO DE 2008. ....	107
TABLA 14: DISTRIBUCIÓN DE USUARIOS EN LOS ENLACES DE INTERNET. ....	108
TABLA 15: DISTRIBUCIÓN DE USUARIOS EN LOS ENLACES DE INTERNET. ....	108
TABLA 16: CAPACIDAD DE CADA ENLACE DE LA RED DE LA UPB SECCIONAL BUCARAMANGA ....	108
TABLA 17: CAPACIDAD DE CADA ENLACE DE LA RED DE LA UPB SECCIONAL BUCARAMANGA. ....	109
TABLA 18: DISTRIBUCIÓN DE LOS ARCHIVOS DE LA MUESTRA SELECCIONADA. ....	116
TABLA 19: TOTAL DE BYTES DE LA RED UPB AL PROXY PANDA EL 18 DE MAYO DE 2006. ....	123
TABLA 20: TOTAL DE BYTES DEL PROXY PANDA ALA RED UPB EL 18 DE MAYO DE 2006.....	124
TABLA 21: TOTAL DE BYTES DE LA RED UPB AL PROXY PANDA EL 19 DE MAYO DE 2006. ....	125
TABLA 22: TOTAL DE BYTES DEL PROXY PANDA ALA RED UPB EL 19 DE MAYO DE 2006.....	126
TABLA 23: TOTAL DE BYTES DE LA RED UPB AL PROXY PANDA EL 20 DE MAYO DE 2006. ....	127
TABLA 24: TOTAL DE BYTES DEL PROXY PANDA ALA RED UPB EL 20 DE MAYO DE 2006.....	128
TABLA 25: CLASIFICACIÓN DE EVENTOS DE ACUERDO AL TAMAÑO DE ARCHIVO. ....	131
TABLA 26: FRECUENCIA, PORCENTAJE Y PORCENTAJE ACUMULADO DE EVENTO. ....	132
TABLA 27: MUESTRA DE LA FECHA Y HORA DE INICIO DE CONEXIÓN. ....	153
TABLA 28: MUESTRA DE LA FECHA Y HORA DEL FIN DE CONEXIÓN. ....	153
TABLA 29: MUESTRA DEL TIEMPO INICIO, TIEMPO FIN Y TIEMPO QUE DURÓ LA CONEXIÓN EN SEGUNDOS. ....	153
TABLA 30: MUESTRA DE LA IP Y PUERTO DE ORIGEN DE LA CONEXIÓN OBTENIDA DE EL ARCHIVO RESULTADOSO.XLS. ....	153
TABLA 31: MUESTRA DE LA IP Y PUERTO DE ORIGEN DE LA CONEXIÓN OBTENIDA DE EL ARCHIVO RESULTADOSI.XLS.....	153
TABLA 32: MUESTRA DE LA IP Y PUERTO DE DESTINO DE LA CONEXIÓN OBTENIDA DE EL ARCHIVO RESULTADOSI.XLS.....	154
TABLA 33: MUESTRA DE LA CANTIDAD DE PAQUETES NECESARIOS PARA TRASMITIR EL ARCHIVO Y EL TAMAÑO DE ARCHIVO.....	154

## LISTA DE ECUACIONES

ECUACIÓN 1 .....	32
ECUACIÓN 2 .....	32
ECUACIÓN 3 .....	32
ECUACIÓN 4 .....	33
ECUACIÓN 5 .....	41
ECUACIÓN 6 .....	41
ECUACIÓN 7 .....	42
ECUACIÓN 8 .....	43
ECUACIÓN 9 .....	44
ECUACIÓN 10 .....	69
ECUACIÓN 11 .....	71
ECUACIÓN 12 .....	71
ECUACIÓN 13 .....	74
ECUACIÓN 14 .....	75
ECUACIÓN 15 .....	75
ECUACIÓN 16 .....	75
ECUACIÓN 17 .....	76
ECUACIÓN 18 .....	77
ECUACIÓN 19 .....	77
ECUACIÓN 20 .....	78
ECUACIÓN 21 .....	78
ECUACIÓN 22 .....	78
ECUACIÓN 23 .....	78
ECUACIÓN 24 .....	79
ECUACIÓN 25 .....	81
ECUACIÓN 26 .....	82
ECUACIÓN 27 .....	83
ECUACIÓN 28 .....	83
ECUACIÓN 29 .....	132
ECUACIÓN 30 .....	133

## RESUMEN GENERAL DE TRABAJO DE GRADO

**TITULO:** EVALUACIÓN DE UN MODELO DE RED DE COLA CERRADA PARA LA ESTIMACIÓN DE LA CAPACIDAD DE CANAL DE UNA RED DE DATOS UNIVERSITARIA

**AUTOR(ES):** ELGA XIMENA CARDOZO MORENO

**FACULTAD:** Facultad de Ingeniería Electrónica

**DIRECTOR(A):** JHON JAIRO PADILLA AGUILAR

### RESUMEN

En el presente trabajo, se usa el modelo de red de cola cerrada (CQN) el cual, es un sistema realimentado de población finita con un único recurso o servidor, que simula el efecto del algoritmo de control de la ventana de congestión de TCP conocido como arranque lento (slow start) y a la vez se usa para calcular la capacidad del enlace para soportar N fuentes ó usuarios a los que se desea garantizar un ancho de banda efectivo. Dado que el tamaño de los archivos varía enormemente, un simple retardo, como por ejemplo 100ms, no es perceptible para todos los archivos, debido a que el tráfico elástico se adapta a la disponibilidad del ancho de banda. Por lo tanto el resultado obtenido depende de la media del tamaño de los archivos transmitidos por el enlace de cuello de botella y el tiempo entre conexiones y no de sus características de autosimilitud. Para conseguir el objetivo, se utiliza una simple expresión tal que el ancho de banda optimo del enlace depende de la cantidad de usuarios de la red, del producto entre tamaño promedio de los archivos transmitidos y la tasa de transferencia, así como del ancho de banda efectivo que se desea garantizar a cada uno. Con el fin de validar los resultados obtenidos y bajo las mismas condiciones de cálculo, se realiza un modelo de simulación de la red y se determina el uso del ancho de banda del enlace.

### PALABRAS CLAVES:

Redes de Cola Cerrada, Ancho de Banda Efectivo, Tráfico Elástico

V° B° DIRECTOR DE TRABAJO DE GRADO

## ABSTRACT

**TITLE:** EVALUATION OF A CLOSED-QUEUING NETWORK MODEL TO ESTIMATE THE EFFECTIVE BANDWIDTH OF THE INTERNET CHANNEL OVER AN ACADEMIC NETWORK

**AUTHOR(S):** ELGA XIMENA CARDOZO MORENO

**FACULTY:** Facultad de Ingeniería Electrónica

**DIRECTOR:** JHON JAIRO PADILLA AGUILAR

## ABSTRACT

In this paper, we use the closed-queueing-network model (CQN), with a finite population and a single server. It simulates the effect of the control algorithm of the TCP congestion window, also known as the "slow start" algorithm. At the same time it is used to calculate the capacity of the link to support N users in order to ensure an effective bandwidth. Since the file size varies greatly, a simple delay, such as 100ms, is not perceptible for all files, because the elastic traffic fits to available bandwidth. Thus, the result depends on the average size of the files transmitted by the bottleneck link and the time between connections and it not depends on their self similar features. To achieve the objective, this work uses a simple expression such that the optimal bandwidth of the link depends on the number of network users, the product between the average size of transferred files and the transfer rate and bandwidth guaranteed to each user. To validate the results, and under the same conditions of theoretical calculation, a simulation model of the network was performed, which showed the bandwidth usage of the link.

## **PALABRAS CLAVES:**

Closed-queueing networks, effective bandwidth, elastic traffic,  
Capacity channel estimation, Internet traffic

**V° B° DIRECTOR DE TRABAJO DE GRADO**

# 1 INTRODUCCIÓN

## 1.1 PLANTEAMIENTO

Actualmente los administradores de red calculan el ancho de banda a partir del ancho de banda que quieren asegurar a cada dispositivo conectado a la red, lo anterior no tiene en cuenta el comportamiento del usuario, ni un análisis del tráfico previo, lo cual puede llegar a crear deficiencias en el cálculo de la misma, logrando así una red congestionada o por el contrario subutilizada [1].

De otra parte las redes IP versión cuatro no pueden garantizar Calidad de Servicio (QoS) para tráfico en tiempo real ya que los recursos no se reservan y todos los paquetes son tratados por igual, de ahí que sea necesario garantizar un ancho de banda para realizar conexiones exitosas y ofrecer a los diferentes usuarios de la red un servicio aceptable [2]

## 1.2 JUSTIFICACIÓN

Este trabajo de grado, se enfoca en estimar la capacidad de canal de una red universitaria y hace parte de un proyecto de investigación de maestría que se realiza desde el año 2005 por el Ingeniero Jaime Alberto Moreno Mogollón sobre el comportamiento de la red de la Universidad Pontificia Bolivariana seccional Bucaramanga, este pretende estimar el comportamiento del tráfico presente en la red, la obtención de parámetros como la capacidad del canal, la tasa de transferencia y la cantidad de usuarios, que permitan llevar a una precisa caracterización de la red con el fin de optimizar los recursos existentes.

A su vez se espera obtener y fortalecer los conocimientos acerca del tráfico de redes para una posterior profundización en el ámbito de la Telemática, modelamiento y simulaciones en Pre-grado (Ingeniería Electrónica) y Post-grado (Especialización en Telecomunicaciones):

La metodología propuesta puede posteriormente ser aplicada por personas interesadas en la administración, diseño y dimensionamiento de redes, que tendrían al alcance un modelo a partir de lo cual podrían tomar decisiones más acertadas sobre el ancho de banda necesario de acceso a Internet.

## **1.3 OBJETIVOS**

### **1.3.1 Objetivo General**

Evaluar el modelo de red de cola cerrada para estimar el ancho de banda necesario de acceso a Internet en una red de datos universitaria y establecer una metodología para su posterior implementación.

### **1.3.2 Objetivos Específicos**

- Ilustrar el funcionamiento de la red de datos de la Universidad Pontificia Bolivariana Seccional Bucaramanga.
- Realizar una búsqueda del estado del arte del modelo de red de cola cerrada.
- Analizar y caracterizar las trazas de tráfico tomadas de la red de datos de la Universidad Pontificia Bolivariana Seccional Bucaramanga para la obtención de las variables necesarias para evaluar el modelo.

- Realizar la búsqueda de un software de simulación que cumpla con las condiciones necesarias para poder realizar una completa evaluación del modelo de red de cola cerrada.
- Implementar el algoritmo asociado al modelo de red de cola cerrada en el software de simulación y calcular los diferentes parámetros de la red.
- Ampliar los estudios que se han realizado con anterioridad sobre el modelo de red de cola cerrada para determinar la capacidad del canal de Internet óptima.
- Elaborar recomendaciones a partir de los resultados observados de acuerdo con las consideraciones administrativas.

#### **1.4 METODOLOGIA UTILIZADA**

Para la realización de este trabajo se tomó en cuenta que principalmente se pretende la evaluación del modelo de red de cola cerrada; [3] para esto, se siguió el siguiente orden metodológico.



En primera instancia se analizó el funcionamiento de la red de datos de la Universidad Pontificia Bolivariana Seccional Bucaramanga, que fue tomada como referencia; para esto fue necesario: (Ver sección 3.2: Descripción de la Operación de la Red)

- Investigar y documentar la estructura y el funcionamiento general de la red de datos de la universidad.
- Investigar y documentar cambios en los recursos de la red que se dieron después de la toma de datos.
- Investigar y documentar restricciones administrativas aplicadas a la red.

Después se investigó y documentó sobre el estado del arte del modelo de red de cola cerrada lo cual implicó: (Ver sección 2.6: Modelo de Red de Cola Cerrada)

- Investigar en la librería digital de la IEEE y descargar los artículos relacionados con el modelo de red de cola cerrada, procesos estocásticos y teoría de colas. Las principales fuentes bibliográficas fueron [1], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].
- Obtener y estudiar trabajos de investigación previos relacionados con la Red de datos UPB seccional Bucaramanga, tales como: [2], [13], [14], [15].

- Consultar y documentar libros de referencia sobre el modelo de red de cola cerrada, procesos estocásticos y teoría de colas, se encontraron: [16], [17], [18], [19], [20], [21], [22], [23].
- Establecer una base de datos con la documentación obtenida.
- Estudiar el modelo de red de cola cerrada y sus respectivos parámetros.

Con el conocimiento de los parámetros necesarios para evaluar el modelo, se analizaron y caracterizaron las trazas de tráfico tomadas de la red de datos de la Universidad Pontificia Bolivariana seccional Bucaramanga en [2], para esto se debió: (ver secciones 2.7 Herramientas de Software Utilizadas y 3.3 Caracterización de las trazas de tráfico)

- Realizar una investigación y documentación sobre la relación entre el comportamiento de las aplicaciones http 1.0 y http 1.1 sobre TCP. Las principales fuentes de información fueron: [24], [25], [26], [27].
- Aplicar a las trazas obtenidas diferentes filtros con el fin de obtener información específica de las conexiones TCP usando la interfaz gráfica de Wireshark. Como manual se tomó: [28]. Software obtenido en: [29].

- Usar el software Honeysnap para extraer las conexiones TCP y UDP. Como manual se tomó de referencia el archivo suministrado por los creadores: [30]. Software obtenido en: [31]
- Usar el editor de Contenidos Sed para filtrar y organizar el contenido de los archivos de texto generados por Honeysnap. Como manual de referencia se tomó: [32]. Software obtenido en: [33].
- Realizar un análisis estadístico usando tablas dinámicas de Excel 2007. Este análisis se enfocó en la utilización del canal de Internet en función de los usuarios y el tamaño de los archivos que transmiten.

A continuación, se realizó la búsqueda del software de simulación que cumpliera con las condiciones necesarias para poder realizar una completa evaluación del modelo de red de cola cerrada; por esto se hizo necesario: (ver secciones: 2.7.5 NS-2 y 3.4 Simulación del comportamiento de la red)

- Realizar un diagrama de la red.
- Recopilar las características del tráfico Web.

- Construcción de la red sobre el software de simulación Network Simulator Ns-2. Tomando como referencia el manual: [34]. Software obtenido en: [35].
- Simular la capacidad del canal de Internet en función de la tasa de transferencia, el número de usuarios y la capacidad entregada a cada uno de manera gráfica.

Después se implementó el algoritmo asociado al modelo de red de cola cerrada usando Visual Basic aplicado para crear una macro en Microsoft Excel 2003 [36], con lo cual se consiguió: (ver sección 3.3.5 Resultados de la utilización del canal de Internet con Excel)

- Evaluar la expresión del ancho de banda efectivo con el criterio de media y obtener la capacidad de canal en la hora de mayor congestión.
- Verificar la validez del modelo tendiendo los resultados empíricos, teóricos y simulados.
- Estimar la capacidad del canal de Internet en función de la tasa de transferencia, el número de usuarios y la capacidad entregada a cada uno.

Con todo lo anterior se amplió el alcance de los estudios que se han realizado con anterioridad sobre el modelo de red de cola cerrada para determinar la capacidad del canal de Internet necesaria y se elaboraron recomendaciones a partir de los resultados obtenidos (ver sección 3.5.2 Resultados obtenidos para la hora pico), además se construyó una metodología (ver sección 3.7 Síntesis de la Metodología Propuesta), con el fin de que el

modelo sea posteriormente utilizado usando software libre o al alcance de un administrador de red o estudiante como Excel 2003 o 2007, (ver sección 2.7 Herramientas de software utilizadas) .

## **2 ESTADO DEL ARTE**

### **2.1 MODELADO Y SIMULACIÓN DE SISTEMAS**

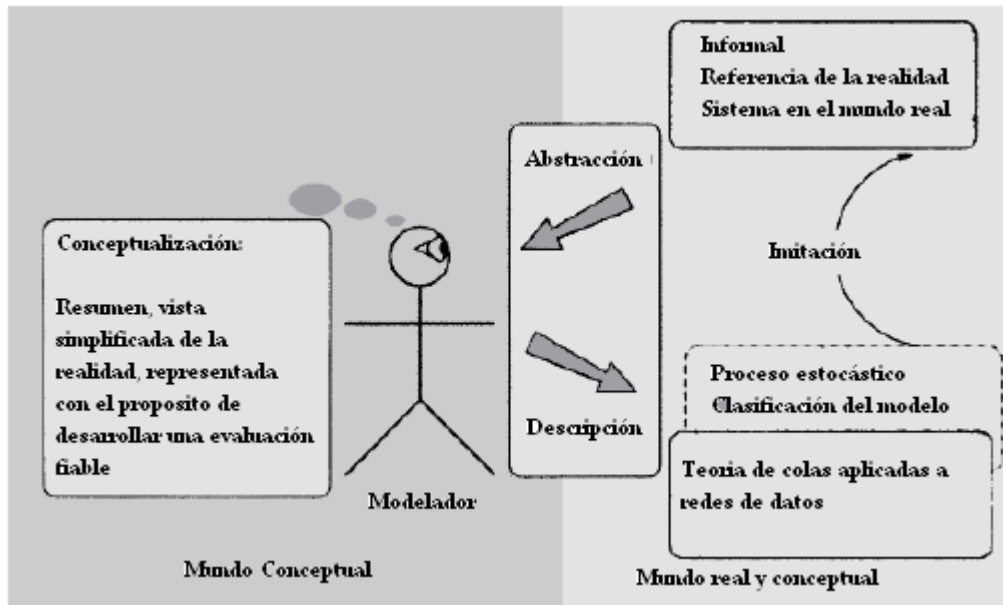
Un modelo es una representación de la realidad, desarrollado con el propósito de estudiarla, por esta razón y teniendo en cuenta que en la mayoría de los análisis no es necesario considerar todos los detalles de la realidad, el modelo no solo es un sustituto de la realidad sino también una simplificación de ella tal como se expresa en [16].

Visto de otra manera la operación de las empresas en la actualidad ha obligado al desarrollo de sistemas complejos cuyos análisis requieren métodos de modelado, con la finalidad de entender, analizar y optimizar su desempeño; el sector de las telecomunicaciones no podía ser ajeno a esta tendencia que gracias al desarrollo de la computación, ha tenido una evolución creciente en los últimos tiempos [17].

Como un ejemplo, si se considera una red de computadores cuyos componentes son, routers, switches, hubs, con líneas de comunicación que satisfacen los requerimientos de transferencia de los paquetes de datos entre varios computadores conectados a la red y, si tenemos como sistema de interés la comunicación de la red solamente, se pueden

reemplazar los computadores por elementos que toman acción sobre la red enviando y recibiendo paquetes únicamente. Muy diferente es el caso del comportamiento del sistema estudiado en el presente trabajo, ya que en un sistema de comunicación como Internet, la carga de trabajo depende de el número de usuarios activos, de tal manera que cuando un usuario específico comienza a acceder a la información de la WWW (World Wide Web) con un navegador, no puede conocerse con anticipación su decisión y por esta razón puede considerarse como un sistema no determinístico o estocástico dado que la estimulación del medio es generalmente impredecible [22].

### 2.1.1 El proceso de modelado



**Figura 1:** Formalización del proceso.  
Tomada de [17] traducido por el autor.

El primer paso para el desarrollo de un modelo de evaluación consiste en la formalización del proceso, durante el cual se genera una descripción formal del sistema en el mundo real como se muestra en la Figura 1.

Para esto usa la conceptualización, esta no es más que un resumen o vista simplificada de la realidad, de esta manera si el proceso esta representado por una cola de red (*queueing network*) el modelador aplica el flujo de trabajo enrutado (*routed job flow*) como paradigma a modelar. El proceso se toma como un conjunto de estaciones de servicio que están conectadas por los bordes a entidades independientes que crean un flujo (*flow*) de datos a través de la red [22].

El segundo paso para el desarrollo de un modelo de evaluación es la deducción de la representación de medidas para la aplicación de los métodos de solución apropiados, dependiendo de la conceptualización escogida durante la formalización del proceso, los siguientes métodos de solución se encuentran disponibles [22]:

- **La solución analítica:** El principio central de la solución analítica esta representado por la descripción formal del proceso como una ecuación donde las medidas interesantes pueden ser obtenidas como soluciones de forma cerrada (*closed-form*), o de un conjunto de sistemas de ecuaciones donde la medida aproximada o exacta puede ser calculada con algoritmos apropiados [22] .
- **Las soluciones de la forma cerrada (*Closed-form*)** están disponibles si el sistema puede ser descrito como una red de colas (*queueing network*) o de la forma producto

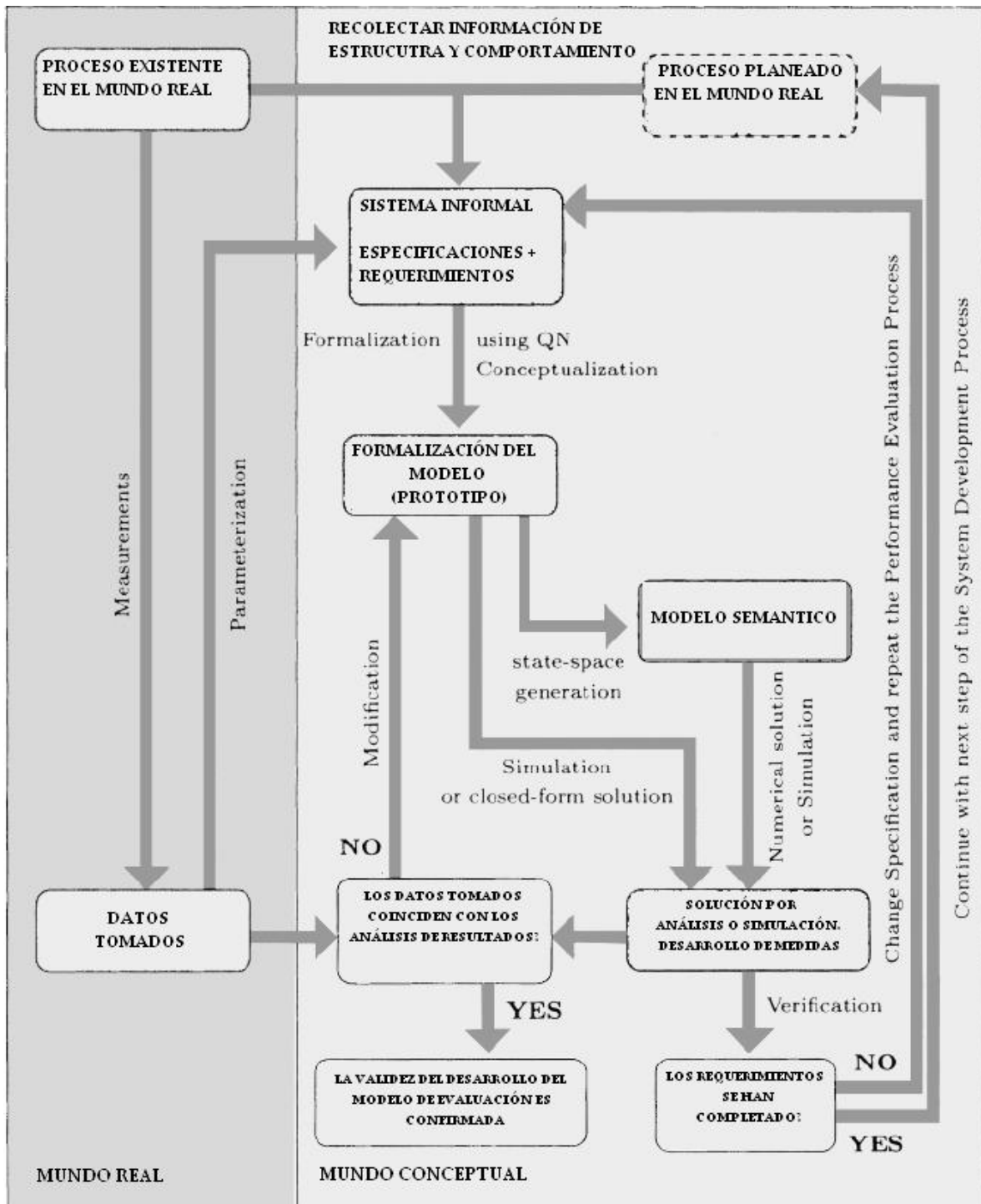


(*product-form*). Esta clase de ecuaciones puede ser derivada de la media del número de trabajos (*jobs*) en las estaciones de servicio. Una gran ventaja de esta forma de solución es su moderada complejidad computacional que permite un rápido desarrollo de medidas incluso para grandes procesos.

- De otra parte **las soluciones numéricas** están compuestas de muchos tipos de ecuaciones que pueden ser derivadas de una descripción formal del proceso que no posee una solución de la forma cerrada (*closed-form*); como por ejemplo en los casos de complejos sistemas integro-diferenciales; en estos casos las soluciones aproximadas pueden obtenerse a través de la aplicación de algoritmos de cálculo numérico. La desventaja con respecto a la solución de la forma cerrada esta en su mayor complejidad computacional [22].

Aparte de la solución analítica también se pueden encontrar las soluciones mediante simulación debido a que no puede deducirse un sistema de ecuaciones o porque la aplicación de algoritmos sea demasiado compleja; a su vez también pueden encontrarse soluciones híbridas que combinan la solución analítica y simulada, entre otras no pertinentes al desarrollo de este trabajo.

A continuación, en la figura 2 se puede observar mediante un diagrama de flujo el proceso anteriormente descrito;



**Figura 2:** Aplicación del desarrollo del modelo de evaluación de un proceso. Tomada de [22] traducido y modificado por el autor.

La primera actividad en el proceso de evaluación es la de recolectar información acerca de la estructura y comportamiento del proceso existente o el proceso planeado, la participación

de alguien experto en este campo es de mucha ayuda y de alguna manera indispensable para aplicaciones complejas. Usualmente recolectar la información es un proceso informal que se documenta de manera textual o gráfica, (en el desarrollo de este trabajo se requiere un proceso metodológico y el uso de software especializado descrito en secciones posteriores). Además, esta descripción informal debe contener la representación, fiabilidad, disponibilidad y confiabilidad que el proceso debe satisfacer. Después de esto se lleva a cabo la formalización del proceso por el modelador quien necesita modelar y aplicar sobre un conocimiento específico [22].

Por otra parte, en [17] se denota que independientemente de la clasificación del modelo, existen ciertas características que este debe tener:

- Confiabilidad
- Sencillez
- Bajo costo de desarrollo y operación
- Manejabilidad
- De fácil entendimiento, tanto del modelo como de los resultados.
- Una relación costo-beneficio positiva.

Como un ejemplo de las características relacionadas anteriormente se podrían resaltar los modelos estocásticos o probabilísticos, que se reconocen porque los valores de sus variables sufren modificaciones aleatorias con respecto a un valor promedio; dichas variaciones pueden ser manejadas mediante distribuciones de probabilidad [16]; lo anterior

hace de estos modelos de gran importancia en el desarrollo de este trabajo ya que un buen número de estos modelos se pueden encontrar en la teoría de líneas de espera; que han sido desarrolladas desde hace varias décadas y que en la actualidad son consideradas la principal herramienta matemática para el análisis del comportamiento y desempeño de las redes de telecomunicaciones, en especial la comunicación de datos [17].

## 2.2 PROCESOS ESTOCÁSTICOS

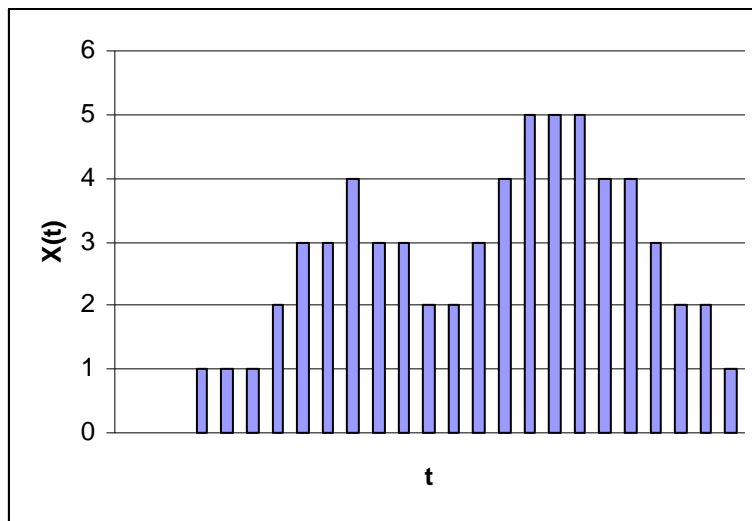
Muchos de los sistemas que encontramos en el mundo real tienen un comportamiento que no podemos describir a través de modelos determinísticos. Un ejemplo es el tráfico de mensajes por una línea de comunicación de datos, el cual es un sistema que varía en el tiempo mientras que pasa por fluctuaciones al azar.[17]

Un proceso estocástico  $X = \{X(t), t \in \mathcal{T}\}$  es un conjunto de variables aleatorias. Es decir para cada  $t$  del conjunto de índices  $\mathcal{T}$ ,  $X(t)$  es una variable aleatoria que toma valores en el conjunto  $\mathcal{S}$ , denominado espacio de estados de proceso [18], es decir donde  $X(t)$  mide, en el instante de tiempo  $t$ , el aspecto del sistema bajo consideración [17].

Suponiendo que el sistema que va a ser descrito es el movimiento de mensajes en un nodo de conmutación en una red y el aspecto que vamos a observar es el número de mensajes que se encuentran en el nodo en un instante de tiempo  $t$ . En cualquier instante de tiempo  $t$ ,

$X(t)$  toma uno de los valores  $0, 1, 2, \dots$  y  $t$  puede ser cualquier valor del conjunto  $(0, \alpha)$ , desde el instante en que se inicio la observación hasta el futuro infinito. No obstante, en la mayoría de los casos este conjunto se restringe al tiempo que dure la observación; en nuestro caso si la observación se realiza durante un día,  $t$  toma valores del conjunto.[17]

Por ejemplo si los mensajes llegan de uno en uno y se observa continuamente el número total de mensajes que se encuentran en el nodo, entonces el valor  $X(t)$ , se incrementa en uno y cuando sale un mensaje disminuye en uno. (Ver figura 3)



**Figura 3:** Número de mensajes,  $X(t)$ , que hay en el nodo en el instante  $t$ . Tomada de [17] modificado por el autor.

Los valores que puede tomar  $X(t)$  son llamados **estados** y el conjunto de los estados es llamado **espacio de estados** [17]. El espacio de estados  $S$ , o lo que es lo mismo, la magnitud medida; puede ser discreta (por ejemplo la longitud de una cola) o continua (por ejemplo, el tiempo de tránsito de un mensaje a través de una red), por lo que habitualmente son denominados, respectivamente, **procesos discretos** (también cadenas) y **procesos**

**continuos** (o, simplemente procesos) [18]. Además  $X(t)$ , puede ser unidimensional, bidimensional, m-dimensional. Adicionalmente, los cambios en el valor de  $X(t)$  reciben el nombre de transiciones entre sus estados [17].

De la misma manera, el conjunto de todos los valores que puede tomar  $t$  es llamado **espacio parametral**,  $\mathcal{T}$  [17]. Por tanto, si el conjunto de índices es discreto, el proceso suele recibir el nombre de secuencia aleatoria y usualmente se denota por  $\{X_n\}$ . Respecto a la connotación temporal, el concepto de tiempo discreto es de interés cuando el tiempo se divide en períodos o ranuras (*slots*) de longitud fija  $T$ . En algunos casos este hecho permite estudiar de forma bastante aproximada y con menor complejidad un proceso continuo. Para este fin, se muestrea el proceso en tiempo continuo de forma que  $\{X_n = X(nT)\}$ . Estos procesos se denominan más específicamente procesos de tiempo discreto para reflejar su connotación temporal [18].

En una familia de variables aleatorias  $X(t)$ , donde  $t$  es un punto en el espacio y  $\mathcal{T}$ , llamado espacio parametral y donde para cada  $t \in \mathcal{T}$ ,  $X(t)$  es un punto en un espacio  $\mathcal{S}$ , llamado espacio de estado. En general es posible tener procesos estocásticos caracterizados por las diferentes combinaciones de espacio de estado y espacio parametral.

### 2.2.1 Espacio Parametral

Es el conjunto de todos los valores que puede tomar el parámetro  $t$ . El espacio parametral puede ser:

- Finito o innumerablemente infinito. Por ejemplo:

$$\mathcal{T} = (\dots, -2, -1, 0, 1, 2, \dots)$$

$$\mathcal{T} = (0, 1, 2, \dots, 100)$$

En este caso se dice que el proceso es de parámetro discreto.

- Continuo. Por ejemplo:

$$\mathcal{T} = \{t / 0 \leq t\}$$

$$\mathcal{T} = \{t / -2 < t < 2\}$$

$$\mathcal{T} = \{t / 4 \leq t \leq 5\}$$

En ese caso se dice que el proceso es de parámetro continuo.

### 2.2.2 Espacio de Estados

Es el conjunto de todos los valores posibles de  $X(t)$ ,  $t \in \mathcal{T}$ . El espacio de estados puede ser:

Continuo o discreto. Según el caso se dice que el proceso estocástico es de espacio de estados continuo o discreto.

Algunas veces deben hacerse suposiciones acerca de un proceso estocástico, que pueden no ser estrictamente ciertas en la práctica, pero sin las cuales el análisis matemático o el tratamiento de estos modelos puede ser demasiado complejo o incluso imposible. Según [18] las mas importantes con:

- Un proceso estocástico  $\{X(t)\}$  es **estacionario en sentido estricto** si para cualquier “u” y cualesquiera “n” instantes de tiempo,  $t_1, t_2, \dots, t_n \in \mathcal{T}$  y  $t_1 + u, t_2 + u, \dots, t_n + u \in \mathcal{T}$ , se cumple que:

$$F_{X(t_1+u), \dots, X(t_n+u)}(x_1, \dots, x_n) = F_{X(t_1), \dots, X(t_n)}(x_1, \dots, x_n)$$

**Ecuación 1**

Es decir, todas las distribuciones conjuntas de dimensión finita del proceso no se ven afectadas por un desplazamiento en el tiempo.

- Otra definición de estacionariedad menos exigente y que goza en la práctica de casi tanta utilidad como la anterior es la del proceso **estacionario en covarianza en sentido amplio o de segundo orden**, que es aquel que verifica:

$$E[X(t)] = \mu \mu \in \mathbb{R} \quad \forall t \in \mathcal{T}$$

**Ecuación 2**

$$\text{Var}[X(t)] = \mu \sigma^2 \in \mathbb{R} \quad \forall t \in \mathcal{T}$$

**Ecuación 3**



Además, la covarianza entre dos variables aleatorias cualesquiera del proceso (llamada en este caso autocovarianza),  $X(t)$  y  $X(t+u)$ , depende única y exclusivamente del desplazamiento  $u$ , y no del instante concreto  $t$ , es decir:

$$\text{Cov}[X(t), X(t+u)] = \gamma(u) \quad \forall t \in \mathcal{T}$$

**Ecuación 4**

Este tipo de procesos estocásticos pueden clasificarse en función del grado de dependencia entre sus variables aleatorias como [18]:

- Procesos estocásticos que exhiben dependencia a corto plazo, cuya función de autocovarianza es integrable en toda la semirrecta positiva real cuando  $\mathcal{T}; \mathbb{R}$ , o es sumable cuando  $\mathcal{T}; \mathbb{N}$ .
- Procesos estocásticos que exhiben dependencia a largo plazo, cuya función de autocovarianza no es integrable en toda la semirrecta positiva real cuando  $\mathcal{T}; \mathbb{R}$ , o no es sumable cuando  $\mathcal{T}; \mathbb{N}$ .

Estos procesos son de utilidad en el estudio de las redes de comunicación actuales, dado que modelan bien procesos estocásticos tales como, por ejemplo, los sucesivos tamaños de cuadros de fuentes de video comprimido, o los sucesivos tiempos entre llegadas consecutivas a un nodo de conmutación en Internet (router) de paquetes procedentes de redes Ethernet [18].

### 2.2.3 Ejemplos de procesos aleatorios

**Ejemplo 1:** En una red de comunicación de datos las estaciones que se encuentran por fuera de la infraestructura de la red introducen paquetes de datos en forma aleatoria. Estos viajan a través de la red hasta salir de ella. En un momento dado es importante conocer cuantos paquetes se encuentran en tránsito dentro de la red, pues de ello depende el retardo que sufre el paquete y a que posiblemente se pueda presentar congestión debido al gran número de ellos en la red. La red debe proceder a controlar la entrada de paquetes hasta que salgan algunos de los que están en tránsito para evitar “trancones” que produzcan retardos demasiado grandes

Por tanto el número de paquetes en la red es una variable aleatoria la cual se observa al transcurrir del tiempo. Así mismo se trata de un proceso estocástico de espacio de estados discreto y parámetro continuo.

**Ejemplo 2:** El retardo en un enlace depende de la capacidad del enlace, de la tasa de paquetes que están interesados en ocupar el enlace y de la longitud del paquete; esta última es una variable aleatoria cuyo valor oscila entre un máximo y un mínimo. Lo mismo ocurre con la tasa de arribo de los paquetes la cual depende del interés de la estación que se encuentra conectada al enlace. Por esto, cuanto mayor sea la tasa de arribo, mayor será el retardo; se encuentran muchas más paquetes compitiendo por un enlace, lo que los obliga a hacer cola. Es importante conocer el número de paquetes que se encuentran en espera para calcular la longitud del buffer que debe albergarlos mientras alcanzan el enlace.

Por tanto, la longitud de la cola puede representarse con un proceso estocástico de espacio de estados discreto y parámetro continuo.

### 2.3 TEORÍA DE LÍNEAS DE ESPERA

La teoría de líneas de espera o también llamada teoría de colas, estudia el comportamiento de sistemas donde existe un conjunto limitado de servidores para atender las peticiones generadas por los usuarios, de tal manera que cuando un usuario envía una tarea al sistema, ésta podrá tener que esperar para ser atendida por algún servidor o incluso, podrá ser rechazada si el sistema no tiene capacidad suficiente para almacenarla en espera de ser atendida [18]. La teoría de colas se ha constituido en la principal herramienta para el análisis de las redes de telecomunicaciones ya que el estudio del desempeño de las redes de comunicación de datos implica el conocimiento del comportamiento de las redes ante situaciones de orden aleatorio, tales como el aumento inesperado del tráfico de paquetes dado el hecho de compartir recursos en las redes de conmutación implica una demanda por ese recurso que es necesario satisfacer.[17]

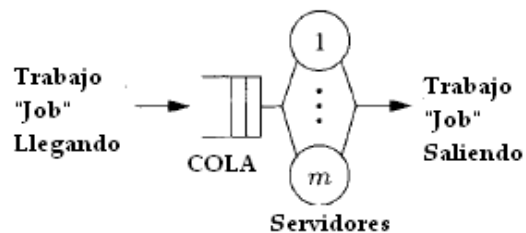
En los modelos de colas existentes, las tareas generadas por los usuarios de una población llegan al sistema en un instante de tiempo aleatorio. Estas tareas demandan una cantidad de servicio también aleatoria, y se sirven una a una o por grupos. El orden en el que se sirven las tareas que ocupan la cola del sistema lo determina una disciplina de gestión de cola; esto implicará el modelado no sólo del sistema en sí, sino también del **comportamiento aleatorio del tráfico ofrecido por los usuarios al sistema** [18].

Una línea de espera puede modelarse como un proceso estocástico en el cual la variable aleatoria se define como el número de transacciones en el sistema en un momento dado; el conjunto de valores que puede tomar dicha variable es  $\{0,1,2,\dots,N\}$  y cada uno de ellos tiene asociado una probabilidad de ocurrencia  $\{P_0,P_1,P_2, \dots, P_N\}$ .

Así mismo en el estudio de sistemas modelados mediante teorías de líneas de espera cuya solución analítica se desconoce, o bien es muy difícil obtener, se suele recurrir a técnicas de simulación que, mediante el empleo de programas de computación, permiten obtener estimaciones de los parámetros de interés del comportamiento del sistema [13].

### 2.3.1 Estructura de una línea de espera.

Una sola estación de un sistema de colas como se muestra en la figura 4 consiste en una cola de tamaño finito o infinito y uno o mas servidores idénticos. Un sistema de cola elemental es también conocido como una estación de servicio o simplemente como un nodo.



**Figura 4:** Estación de servicio con M servidores.  
Tomado de [22], traducido y modificado por el autor.

Una línea de espera se representa mediante la llegada de transacciones a un sistema con el fin de recibir un servicio por una o más entidades dispuestas para ello, conocidas como servidores. Un servidor solo puede atender a un cliente al mismo tiempo, por lo tanto solo tiene dos estados, ocupado (*Busy*) o libre (*Idle*) [22]. En caso de que todas las entidades se encuentren ocupadas, la transacción permanece en espera en la fila hasta que decide abandonar la fila sin ser atendido, o bien, es seleccionado de acuerdo con cierta regla para recibir atención. Una vez el servicio ha sido completamente proporcionado, la transacción sale del sistema y se convierte de nuevo en una transacción potencial [16].

A continuación se describirán cada uno de los elementos que conforman una cola:

- **Servidores:** Representan al mecanismo por el cual las transacciones reciben de una manera completa el servicio deseado. Estas entidades se encuentran dispuestas en forma paralela a la cola o fila, de tal manera que las transacciones pueden seleccionar a cualquiera de ellas para el suministro de dicho servicio. Las dos características principales de los servidores son: la cantidad asignada en cada fila existente en el sistema, la distribución de probabilidad del tiempo de atención a las transacciones o de la velocidad de servicio; dentro de las distribuciones más comunes están la exponencial, la Erlang, la hiperexponencial [16].

- **Transacciones potenciales:** Representan el número total de clientes que podrían requerir el servicio proporcionado por el sistema. Es necesario definir dos características para este conjunto de elementos; la primera tiene que ver con el tamaño del conjunto potencial de clientes, estos pueden ser conjuntos limitados o finitos y en otros casos conjuntos ilimitados o infinitos [16]. En las redes de computadores, por supuesto, todas las estaciones de trabajo, se consideran como elementos de la población, es importante que quede claro que son todos los elementos que “potencialmente requieren del servicio” [17]. La segunda característica se refiere a la distribución de probabilidad del tiempo entre llegadas, o bien a la tasa de entrada promedio [16]. Por ejemplo, en redes de comunicación de datos es muy común utilizar el bit/segundo o bps como unidad para medir la tasa de entrada de elementos al sistema. También se utiliza paquetes/segundo [17].

- **Fila:** Es el conjunto de transacciones que espera ser atendido por alguno de los servidores del sistema. Una fila tiene tres características principales; la primera se refiere a la capacidad, o sea, al número máximo de transacciones que pueden permanecer en ella en un mismo instante y de acuerdo con este número se clasifican como finitas o infinitas. Hay que hacer notar que en el caso de los modelos con tamaño finito, la solución es mucho más fácil de encontrar a partir de las ecuaciones generales ya que la solución del modelo se reduce a un sistema de ecuaciones simultaneas y a la evaluación de las medidas de desempeño mediante promedios ponderados, mientras que, en el caso de modelos de tipo limitado o infinito, es necesario recurrir a la solución del sistema de ecuaciones así como a

la evaluación de las medidas de desempeño y a algunas series geométricas que dificultan en cierto grado el manejo algebraico de la solución. La segunda característica es el orden en que las transacciones son extraídas de la fila para su atención, en ese caso podemos encontrar: primero en llegar/primero en salir (*FIFO- First in/First out*), por prioridad (*PQ- Priority Queuing*), orden aleatorio, etc, y por último la forma de salir de la fila, que puede darse mediante el proceso de servicio o bien, mediante el abandono por diversos factores [16].

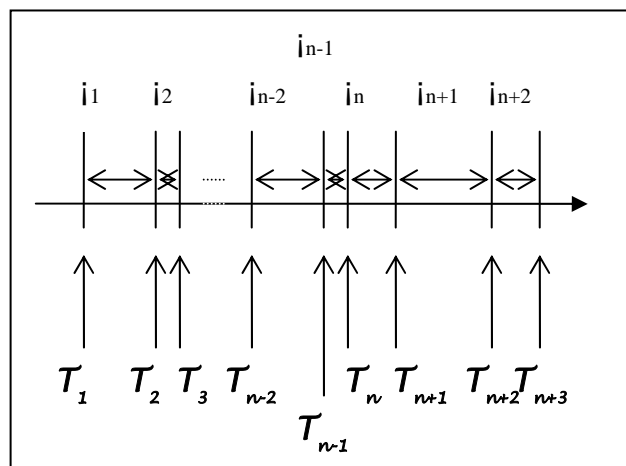
### 2.3.2 Patrón de llegadas

Las sucesivas peticiones de servicio generadas por los usuarios llegarán al sistema en instantes impredecibles [18]. Este comportamiento se modelará de dos formas distintas en función de la dependencia o independencia de las llegadas respecto al estado del sistema:

- **Llegadas Independientes:** Esta situación podrá darse, o bien cuando el usuario o usuarios generen tareas sin esperar a que éstas terminen de ser servidas, o bien cuando aun esperando un usuario a que termine de ser atendida su tarea antes de generar otra, el número de usuarios sea elevado y pueda considerarse despreciable la diferencia entre las velocidades de llegada de tareas al sistema en función de su estado. En este caso se modelará de forma conjunta el comportamiento de todas las llegadas al sistema [18].

- Llegadas Dependientes:** Esta situación se dará cuando, por ejemplo, esperando el usuario a la atención antes de una nueva generación, el número de usuarios actualmente inactivos tenga un impacto palpable en la velocidad a la que llegan las tareas al sistema: a más usuarios inactivos (menos tareas de usuarios en el sistema), mayor velocidad de servicio de tareas. En este caso se modelará el comportamiento individual de los usuarios, caracterizando el tiempo entre que un usuario queda inactivo y genera una nueva petición, y considerando estos tiempos, tanto de un mismo usuario como de varios usuarios, independientes entre sí e idénticamente distribuidos [18].

En el modelado del comportamiento estocástico de las llegadas en el caso de independencia, están dados los sucesivos tiempos aleatorios de llegada de las tareas, tal y como se representa con la figura 3.2, éstos componen el proceso estocástico continuo de tiempo (índice) discreto  $\{T_i \in \mathbb{R}^+; i = 1, 2, \dots\}$ . A partir de él, se definen los dos procesos estocásticos utilizados para la caracterización de las llegadas al sistema:



**Figura 5:** Llegada de tareas al sistema.  
Tomado de [18]



- **Proceso de llegadas:** Se define como el proceso estocástico discreto de tiempo continuo que representa, para cada instante de tiempo, el número de tareas totales que llegan al sistema; es decir,

$$A \triangleq \{ A(t) = \arg \max_i (T_i < t) \in \mathbb{N}^*; t \in \mathbb{R}_+ \}$$

**Ecuación 5**

- **Proceso de tiempo entre llegadas:** Se define como el proceso estocástico continuo de tiempo (índice) discreto que representa los tiempos entre llegadas consecutivas; es decir,

$$Y \triangleq \{ Y_i \triangleq T_{i+1} - T_i; i = 1, 2, \dots \}$$

**Ecuación 6**

Usualmente se modela el proceso de tiempo entre llegadas,  $Y$ , como un proceso estocástico estacionario en covarianza, de media  $\overline{Y}$  y con autocorrelación debida a la posible dependencia de la generación de sucesivas peticiones por parte de un usuario o entre las generaciones de distintos usuarios entre sí. En el caso de no existir esta dependencia, se modelará  $Y$  como un proceso estocástico aleatorio puro, es decir, con sus variables aleatorias independientes e idénticamente distribuidas (IID), en este caso nos referiremos a  $Y$  simplemente, sin pérdida de información, como variable aleatoria de tiempo entre llegadas (consecutivas) de peticiones (tareas) al sistema.

Un parámetro característico del comportamiento de las llegadas, será su velocidad media o tasa de llegada, que vendrá dada por:

$$\lambda = 1/E[Y] = 1/\bar{Y}$$

**Ecuación 7**

Donde el operador estadístico  $E[Y]$  representa el valor medio de  $Y$ .

### **2.3.3 Demanda y mecanismo de servicio**

Una tarea que llega al sistema tardará una cantidad de tiempo impredecible en ser servida por uno de los recursos. Para modelar la sucesión de estos tiempos para las sucesivas tareas llegadas al sistema, se usará la notación del proceso de tiempo de servicio demandado  $S \triangleq \{S_i; i=1,2,\dots\}$ , estos tiempos podrán ser dependientes o independientes del sistema según [18]:

- **Tiempos de Servicio Independientes:** Ocurren cuando la configuración del sistema no tiene ninguna influencia en el tiempo demandado por las tareas, como por ejemplo con las llamadas telefónicas donde modificaciones en la red telefónica no pueden influir en la mayor o menor duración de las llamadas
- **Tiempos de Servicio Dependientes:** Ocurren cuando la configuración del sistema puede determinar un mayor o menor tiempo de servicio de las tareas, como por ejemplo, en

la transmisión de un paquete de datos el tiempo de transmisión depende de la velocidad de transmisión de la línea.

Para este segundo caso se definen:

- Proceso de cantidad de servicio demandado (L): Este se define como  $L \triangleq \{ L_i ; i=1,2,\dots \}$ . Esta demanda de las tareas de los usuarios medidas en unidades de servicio por ejemplo bits, paquetes, etc.
- Capacidad de los recursos (C): Es el número de unidades de servicio atendidas por cualquiera de los recursos (servidores) por unidad de tiempo. Las unidades serán por ejemplo bits/s, paquetes/s, etc.

En consecuencia, se puede obtener la relación de las variables de los procesos de Tiempo de Servicio Demandado y Cantidad de Servicio Demandado:

$$S_i = L_i / C; i \in N$$

**Ecuación 8**

En el caso de independencia entre S y el sistema (Teniendo en cuenta que C=1), se puede definir a L como variable aleatoria Cantidad de Servicio Demandado, en consecuencia se puede definir a S, como variable aleatoria tiempo de servicio Demandado [18].

Es importante tener en cuenta que el Tiempo de Servicio es una variable aleatoria continua que depende del tipo de transacción que vaya a realizar el Usuario. En las redes de conmutación de paquetes depende de la Velocidad del Enlace (C) y la Longitud del Mensaje que se transmite. Por lo general los mensajes son de longitud variable y responden a una distribución de probabilidad que puede ser conocida si observamos continuamente los mensajes que llegan a un nodo determinado de la red. De esta manera la distribución de probabilidad del Tiempo de Servicio es igual a la distribución de la longitud del mensaje [17].

Para la caracterización del mecanismo de servicio, se define el parámetro Velocidad Media de servicio como:

$$\mu = 1/S$$

**Ecuación 9**

Que representa la máxima tasa a la que un recurso puede atender tareas, es decir, la tasa de atención si el recurso estuviese ocupado el 100% del tiempo.

### **2.3.4 Disciplina de servicio**

Una vez ingresan los elementos al sistema quedan sometidos a una Disciplina de Servicio impuesta por ésta. La Disciplina de Servicio es la forma como el sistema presta el servicio, es decir, el orden en que este atiende los elementos, o descrito de otra manera, La

Disciplina de Servicio es la encargada de establecer el modo de acceso a los recursos del sistema por las tareas que están esperando en la cola [17]. La disciplina mas común es la FCFS (First Come First Served), también llamada FIFO (First In First Out), la cual sirve a las tareas en el orden de llegada al sistema. No siempre es la más conveniente, ya que si hay una tarea muy larga, esta monopolizará el uso del recurso que ocupe [18]. También, entre muchas otras, se encuentra la disciplina LCFS (Last Come First Served) ó LIFO (Last In First Out) donde la última tarea en llegar es la primera en servirse [18].

La atención Aleatoria no considera el orden de llegada y solo se limita a prestar el servicio a cualquiera de los elementos que se encuentran en el sistema en un momento determinado.

La disciplina de servicio también puede darse por prioridades. La prioridad es asignada a los elementos de la población y es considerada en el momento de la atención. Esta prioridad tiene dos formas de prestarse de acuerdo con una definición previa hecha por el sistema:

- **Prioridad con Truncamiento o con apropiación (Preemptive):** Esta implica la suspensión inmediata del servicio cuando llega un elemento con mayor prioridad [17]. De la misma manera la tarea desplazada no pierde el trabajo realizado hasta ese momento si la prioridad con truncamiento es sin pérdidas (Resume); por otra parte, cuando el truncamiento es con pérdidas (No Resume) la tarea desplazada no almacena la información necesaria para poder continuar en el futuro con su trabajo en el punto donde lo dejó [18].

- **Prioridad sin Truncamiento o sin apropiación** (No Preemptive): En este caso, aunque se produzca la llegada de una tarea más prioritaria al sistema que alguna que ya está disfrutando del servicio, esta deberá esperar a que termine de servirse la tarea que se encuentra en el sistema para disfrutar de un recurso.

El manejo de prioridades es de mucho cuidado y requiere una política clara de asignación para evitar que un elemento de menor prioridad se quede en forma permanente en la cola esperando a que lo atiendan.

### 2.3.5 Definiciones y Notación

En 1953 Kendall y Lee propusieron un sistema de clasificación de los sistemas de líneas de espera, ampliamente utilizado en la actualidad [16]. A esta notación se la conoce como “Notación Kendall”, la cual considera seis de las características más importantes de una cola, desde la distribución de probabilidad de entrada hasta la disciplina de servicio [17]; expresándolas en el formato  $(A / B / m / K / N / Z)$ , donde :

- **A:** Es la distribución de probabilidad de entrada. Identifica una variable aleatoria discreta que mide el número de elementos que ingresan al sistema en un intervalo de tiempo. Se trata de una función de conteo. Se utiliza la letra **M** (Markov) cuando la distribución de entrada es Poisson, la letra **G** cuando es una distribución general, **E<sub>k</sub>** cuando es Erlang, **D** cuando es determinística [17].


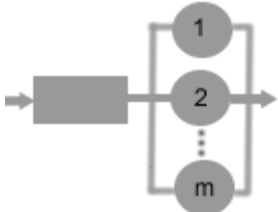
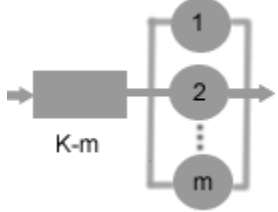
- **B:** Equivale a la distribución de probabilidad del tiempo de servicio. Identifica una variable aleatoria continua que mide el tiempo que dura el servicio. Se utiliza la letra **M** (Markov) cuando la distribución es exponencial, la cual esta asociada a un proceso de Poisson, la letra **G** para cualquier tipo de distribución y la **D** para determinística [17].

También se pueden observar los siguientes símbolos en normalmente usados en las posiciones “A y B”: **H<sub>k</sub>**, que representa la distribución hiperexponencial con parámetro  $k$ , **GI** es la distribución general Independiente y **C<sub>k</sub>** es la distribución Cox con parámetro  $k$  [22].

- **m:** Es el número de servidores en paralelo. Cuando la disposición sea otra, se debe señalar explícitamente, pues la notación Kendall no lo permite [17].
- **K:** Representa la capacidad del sistema o el número de elementos máximo que admite el sistema en servicio y en cola. Cuando la capacidad es finita se indica el número máximo de elementos que permite el sistema y cuando la capacidad es infinita se deja en blanco esta casilla. En la práctica una capacidad grande en comparación con el número promedio de elementos en el sistema se puede considerar como infinita [17].
- **N:** Determina el tamaño de la población. Es el número de elementos que potencialmente requieren del servicio que presta el sistema. Cuando el tamaño es infinito o muy grande en comparación con la capacidad del sistema no se indica en al casilla; ésta se deja en blanco [17].

- **Z:** Indica la disciplina de servicio. Cuando la disciplina es FIFO, se deja la casilla en blanco. En otro caso se indica la disciplina de servicio [17].

Algunos ejemplos de este tipo de notación se observan en la tabla 1. De la misma manera es importante resaltar que generalmente solo se encuentran señalados los apartes **1/2/3** como lo señala el autor en [22] y que puede encontrarse eventualmente la notación Kendall extendida como se describió en un principio.

Notación Kendall	Características de la Cola	Diagrama
M/M/1	<p>Es la más sencilla de todas, pero la más importante, ya que reúne las características de un servicio aleatorio en extremo y un arribo totalmente independiente.</p> <p>La notación nos indica que el arribo es Poisson, el servicio es exponencial, un servidor, capacidad infinita, población infinita y disciplina FIFO [17].</p>	
M/M/m	<p>Entrada Poisson, servicio exponencial, m servidores, capacidad y población infinitas y disciplina FIFO.</p> <p>En este caso tenemos una tasa de salida variable, dependiente del estado del sistema. Este aumento sigue hasta que todos los servidores se ocupan, de allí en adelante la tasa permanece constante y los elementos adicionales que lleguen pasan la cola [17].</p>	
M/M/m/K	<p>Un sistema de este tipo tiene una capacidad limitada a K tareas, con m servidores para atender las tareas que entran, y se considera que la población es finita [18]</p>	

**Tabla 1:** Ejemplos Notación Kendall.  
Tomado de [13].



### 2.3.6 Fórmula de Little

Las llamadas fórmulas de Little facilitan el cálculo de los parámetros de cola tales como; la longitud de la cola, longitud del sistema, tiempo en la cola y tiempo en el sistema [17]. De esta manera; las fórmulas de Little ofrecen las siguientes relaciones entre estas variables:

- La Longitud Promedio de la cola = Tasa promedio de arribo \* tiempo promedio en la cola
- La Longitud Promedio del sistema = Tasa promedio de arribo \* tiempo en el sistema.
- Tiempo en el sistema = Tiempo en la cola + Tiempo de servicio.

Se hace necesario notar que, de acuerdo al autor que se este estudiando estos parámetros pueden denotarse de diferente manera y lo primordial en estos casos e tener claridad sobre los conceptos antes relacionados. Los parámetros descritos por las fórmulas de Little son:

- **La Longitud Promedio de la cola:** Es número promedio de clientes en la fila, es decir, es el promedio que se obtiene solamente del número de clientes que se encuentran en espera de ser atendidos en cualquier momento[16].
- **La tasa promedio de arribo o tasa de entrada promedio:** Es el valor ponderado de las tasas de entrada a un sistema y representan el número promedio de clientes que, efectivamente, ingresan al sistema convirtiéndose de clientes potenciales en clientes reales.

- **El tiempo promedio en la cola o tiempo promedio de espera en la fila:** Se entiende como el promedio de los tiempos de permanencia de los clientes en espera de ser atendidos y se contabiliza desde el momento en que un cliente se une a la fila hasta que cualquiera de los servidores comienza a atenderlo [16].
- **Longitud promedio del sistema o número promedio de clientes en el sistema:** Es el promedio ponderado de los diferentes estados del sistema, definiendo el estado del sistema como el número de clientes que se encuentran acumulados tanto en espera como en servicio en cualquier momento.
- **El tiempo en el sistema o tiempo promedio de espera en el sistema:** Es el promedio de los tiempo de estancia de los clientes y se contabiliza desde el punto en el tiempos en que un cliente entra a la fila hasta el momento en que termina de ser atendido por el servidor.

### **2.3.7 Medidas orientadas al gestor del sistema**

Estas medidas están orientadas al administrador del sistema, puesto que le servirán tanto para conocer, cuál es la demanda de recursos realizada por los usuarios con el fin de planificar el sistema.

- **Intensidad de tráfico (I):** Se define como el número medio de recursos ocupados de un sistema de referencia capaz de atender a todas las peticiones de los usuarios con recursos de igual capacidad que el sistema real.

Cuando el tráfico ofrecido por los usuarios se modele mediante población finita, entonces el sistema de referencia se elige tal que tenga los recursos suficientes para atender simultáneamente todas las posibles peticiones, es decir,  $m=N$  en  $M/M/m/N$ .

- **Caudal (Throughput,  $\bar{\lambda}$ ):** Establece una medida de la Productividad del Sistema, y se define como el número medio de tareas servidas por unidad de tiempo.
- **Factor de utilización ( $\rho$ ):** Mide la probabilidad de Ocupación de un recurso del sistema, es decir, indicará el tanto por ciento del tiempo en que dicho recurso tiende a estar ocupado.

Además de conseguir un caudal de salida alto, al proveedor del servicio también le interesará un alto aprovechamiento de la potencia de los recursos desplegada, por tanto, ambos serán usualmente objetivos contrapuestos [18].

### 2.3.8 Medidas orientadas al usuario

Este grupo de medidas son la que medirán el grado de calidad de servicio que percibe el usuario. Esta calidad será percibida en magnitudes tales como el tiempo que tarda en ser

atendido, el número de peticiones atendidas antes que la suya, o en la posibilidad de que su petición sea rechazada. Las más importantes son el Tiempo de Espera en la cola ( $W_i$ ) y el Tiempo de Respuesta (tiempo transcurrido desde que dicha tarea entra en el sistema hasta que lo abandona,  $T_i$ ).

## **2.4 DESCRIPCIÓN DEL PROTOCOLO TCP**

### **2.4.1 Objetivos de TCP**

TCP cuenta con tres objetivos principales [37]:

- Adaptar la velocidad de transmisión de paquetes al ancho de banda disponible.
- Evitar la congestión de la red.
- Crear una conexión confiable para retransmitir los paquetes perdidos.

### **2.4.2 Cabecera TCP**

La entidad TCP transmisora y la receptora intercambian datos en forma de segmentos. Un segmento consiste en una cabecera TCP fija de 20 bytes (más una parte opcional), seguida de cero o más bytes de datos. El software de TCP decide el tamaño de los segmentos,

teniendo en cuenta dos límites que restringen su tamaño: El primero consiste en que cada segmento, incluida la cabecera TCP, no puede superar la carga útil de 65.535 bytes del IP, mientras que el segundo límite se basa en que cada red tiene una unidad máxima de transferencia o MTU y cada segmento no puede superar este valor. En el caso particular de las redes Ethernet es de 1500 bytes.

<b>Puerto de Origen</b>				<b>Puerto de Destino</b>				
<b>Número de Secuencia</b>								
<b>Número de Reconocimiento</b>								
data offset	Reserved	URG	ACK	PSH	RST	SYN	FIN	<b>Tamaño de la Ventana</b>
<b>Suma de Comprobación</b>				<b>Apuntador Urgente</b>				
<b>Opciones ( 0 o más palabras de 32 bits)</b>								
<b>Datos ( Opcional )</b>								

**Figura 6** Cabecera TCP.  
Tomado de [26] y modificado por el autor.

La figura 6 muestra la distribución de cabecera de un segmento TCP. Es importante resaltar que los segmentos sin datos se usan por lo general para acuses de recibo y mensajes de control.

Los campos de puerto de origen y puerto de destino identifican los extremos locales de la conexión. Cada Host puede decidir por sí mismo la manera de asignar sus propios puertos comenzando por el 256. La dirección de un puerto mas la dirección IP de el Host forman un punto de acceso al servicio de transporte (TSAP) **único** de 48 bits [38].

Cada byte de una conexión TCP tiene su propio número de secuencia de 32 bits. Se usan los números de secuencia tanto para los acuses de recibo como para el mecanismo de ventana de congestión [38].

La longitud de cabecera TCP identificada como “*data Offset*” indica la cantidad de palabras de 32 bits contenidas en la cabecera [38].

A continuación se encuentra un campo de 6 bits identificado como “*reserved*” el cual no se usa y esta destinado para la corrección de errores en el diseño original de TCP [38].

La cabecera TCP también contiene seis banderas de un bit de longitud [38]:

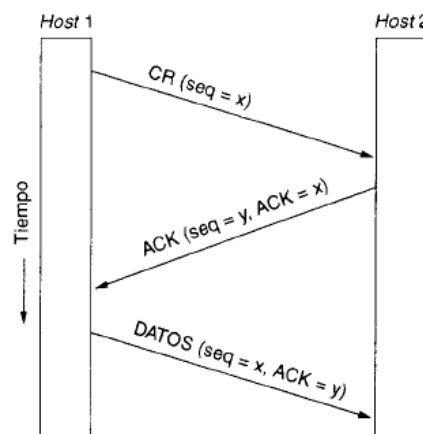
- URG se establece en 1 si esta en uso el apuntador urgente.
- ACK se establece en 1 para indicar que el número de acuse de recibo es válido.
- PSH indica al receptor que debe entregar los datos a la aplicación a su llegada.
- RST se usa para reestablecer una conexión que se ha perdido.
- SYN se usa para establecer conexiones.
- FIN se usa para liberar una conexión.

El control de flujo en TCP se maneja usando una ventana deslizante de tamaño variable. EL campo “Tamaño de ventana” indica la cantidad de bytes que pueden enviarse comenzando por el byte que ya se ha enviado acuse de recibo.

El campo de opciones se diseñó para contar con una manera de agregar características extras no cubiertas por la cabecera normal.

### 2.2.2 Conexiones TCP

TCP establece conexiones usando el protocolo de acuerdo de tres vías desarrollado por Tomlinson en 1975. (Ver figura 7)

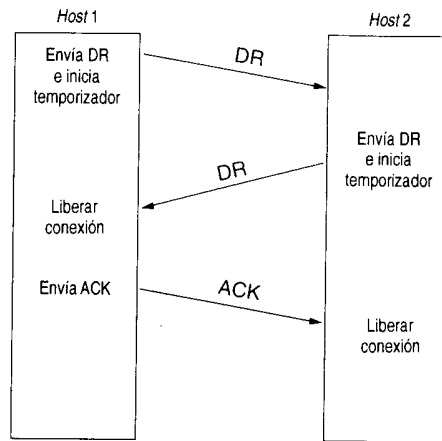


**Figura 7:** Establecer una conexión mediante el protocolo de acuerdo de tres vías.  
Tomado de [38].

El Host 1 escoge un número de secuencia cualquiera y envía al Host 2 una unidad de dato de protocolo de transporte conocida como TPDU que lo contiene. El Host 2 responde con

una TPDU reconociendo el número de secuencia del Host 1 y anunciando su propio número de secuencia inicial. Por último el Host 1 reconoce la selección de un número de secuencia inicial del Host 2 en la primera TPDU de datos que envía.

Como se observa en la figura 8, para liberar una conexión TCP, el Host 1 envía una solicitud de desconexión (*Disconnection request, DR*). Suponiendo que no se pierden mensajes, el Host 2, recibe la solicitud y devuelve una TPDU solicitando la desconexión e inicia un temporizador en el caso de que se pierda su DR. Al llegar esta DR al Host 1, este envía de regreso una TPDU y libera la conexión. De la misma manera, cuando el TPDU llega al Host 2, este libera la conexión.



**Figura 8:** Liberar una conexión mediante el protocolo de acuerdo de tres vías.  
Tomado de [38].



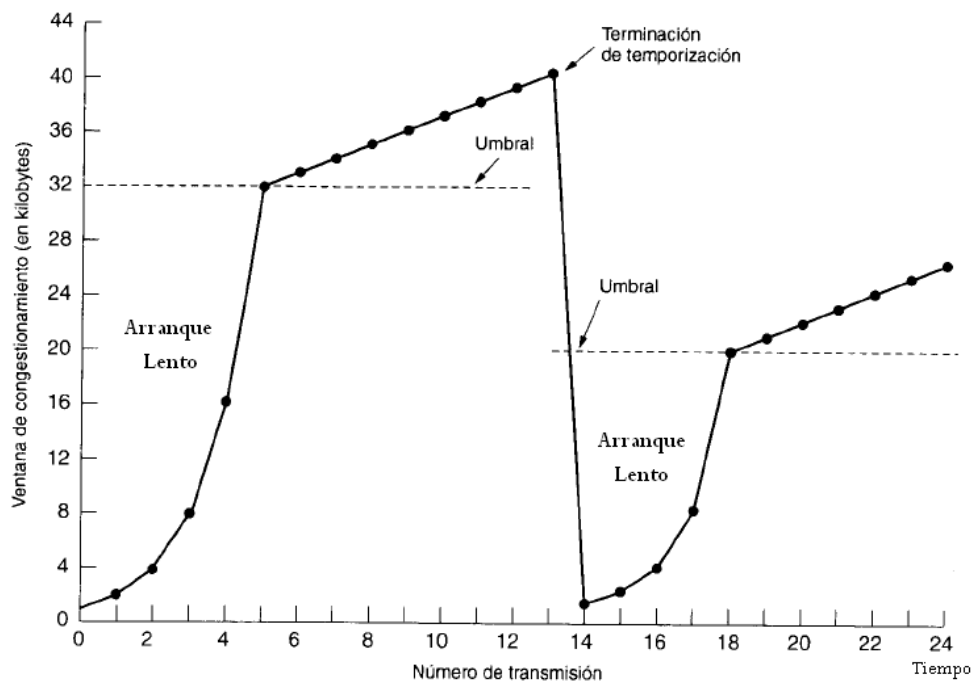
### 2.4.3 Control de Congestión en TCP

En teoría puede manejarse la congestión aprovechando un principio de la física: la ley de conservación de los paquetes, lo que significa que para cualquier tiempo dado, la integral de la densidad de paquetes del bucle entre transmisor-receptor-transmisor es una constante [27].

La solución de Internet es aceptar que existen dos problemas potenciales para que no se cumpla con la ley de conservación de paquetes: la capacidad de la red y la capacidad del receptor [38]. Para ello, al establecerse una conexión, se tiene que seleccionar un tamaño de ventana adecuado. El receptor puede especificar una ventana con base en su tamaño de *buffer*. Si el transmisor se ajusta a su tamaño de ventana, no ocurrirán problemas por desbordamiento de *buffer* en la terminal receptora, pero aún pueden ocurrir debido a congestionamientos internos de la red; por tanto cada transmisor mantiene dos ventanas: la ventana que ha otorgado el receptor y una segunda ventana; la ventana de congestionamiento; la cantidad de bytes que pueden enviarse es al cifra menor de las dos ventanas [38]. Para hacer uso de la ventana de congestión se creó el algoritmo de arranque lento de TCP (*Slow Start*) por Jacobson en 1988 [27].

- **Algoritmo de Arranque lento (*Slow Start*):** Al establecerse una conexión, el transmisor asigna a la ventana de congestión el tamaño de segmento máximo usado por la conexión; entonces envía un segmento máximo. Si se recibe acuse de recibo de este segmento antes de terminar el temporizador, el transmisor agrega el equivalente en bytes de

un segmento a la ventana de congestión para hacerla de dos segmentos de tamaño máximo y envía dos segmentos. Cuando la ventana de congestión es de  $n$  segmentos, si de todos los  $n$  se reciben acuses de recibo a tiempo, se aumenta el tamaño de la ventana de congestión en la cuenta de bytes correspondiente a  $n$  segmentos. De hecho, cada ráfaga reconocida con éxito duplica la ventana de congestión. La ventana de congestión sigue creciendo exponencialmente hasta ocurrir una terminación de temporización o alcanzar el tamaño de la ventana receptora como se observa en la figura 9; la idea es que, si las ráfagas de 1024, 2048 y 4096 bytes funcionan bien, pero una ráfaga de 8192 produce una terminación de temporización la ventana de congestión debe establecerse en 4096 bytes [38].

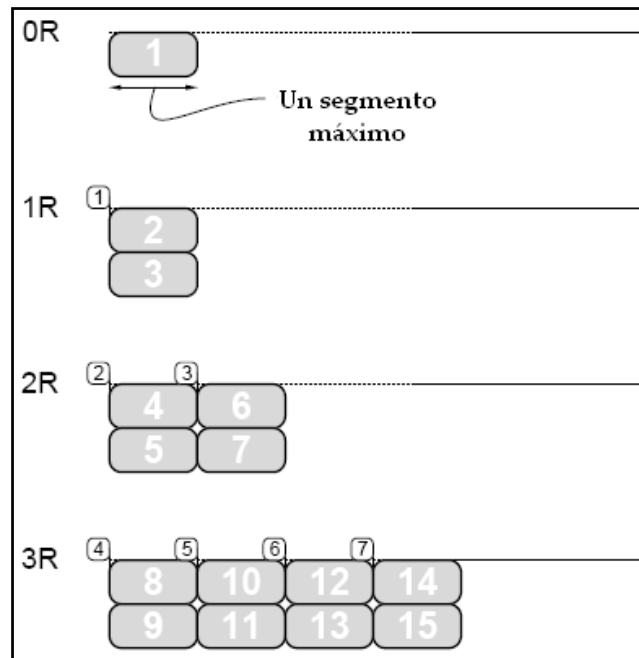


**Figura 9:** Algoritmo de Congestión en Internet  
Tomado de [27] y modificado por el autor.

El algoritmo de control de congestiónamiento de Internet, usa un tercer parámetro, el umbral, el cual se establece inicialmente en 64K, además de las ventanas de recepción y de congestión. AL ocurrir una terminación de temporización, se establece el umbral en la mitad de la ventana de congestiónamiento actual, y la ventana de congestiónamiento se restablece a un segmento máximo como se observa en la figura 9. Luego se usa el arranque lento (slow start), para determinar lo que puede manejar la red. Excepto que el crecimiento exponencial termina al alcanzar el umbral. Si no ocurren más terminaciones de temporización, la ventana de congestiónamiento continuará creciendo hasta el tamaño de la ventana del receptor. En ese punto, dejará de crecer y permanecerá constante mientras no ocurran más terminaciones de temporización y la ventana del receptor no cambie de tamaño, como se observa en la figura 9 [38].

#### **2.4.4 Tránsito de archivos con TCP**

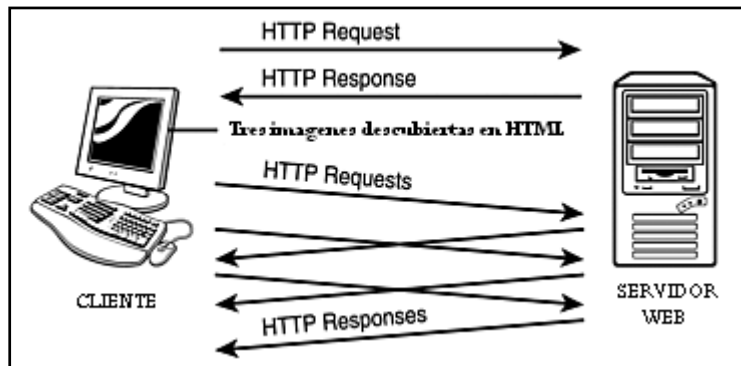
La mayoría del tráfico sobre Internet se constituye en la transferencia de archivos. La media de tamaño de archivo transmitido está cerca de los 10K bytes [37], [39]. Esto significa que la media de tamaño de archivo no requiere más de 10 paquetes TCP tomando en cuenta que un paquete de TCP típico es de 1K byte de tamaño [40], [41]. En consecuencia la mayoría de la transferencia de archivos termina en la tercera fase del algoritmo de arranque lento como se observa en la figura 10.



**Figura 10:** Fases del Algoritmo de arranque lento (Slow Start)  
Tomado de [38] y modificado por el autor.

## 2.5 DESCRIPCIÓN DEL PROTOCOLO HTTP

Los archivos extraídos de Internet son principalmente páginas HTML. Una página HTML que contiene imágenes es una combinación de múltiples objetos, lo que genera a su vez múltiples peticiones a un servidor [42]. Como puede observarse en la figura 11, en términos de rendimiento del sistema es importante entender que un simple “clic” dado por el cliente puede generar una serie de peticiones de archivos en el servidor.



**Figura 11:** Página HTML con tres imágenes.  
Tomada de [25] y modificada por el autor.

El software que principalmente interviene en los servidores Web, es el software que se encarga de gestionar el protocolo HTTP. Este un protocolo de nivel de aplicación que es usado para comunicaciones entre clientes y servidores Web [38]. Del lado del usuario, se utiliza un navegador tal como el Internet Explorer, el mozilla, etc.

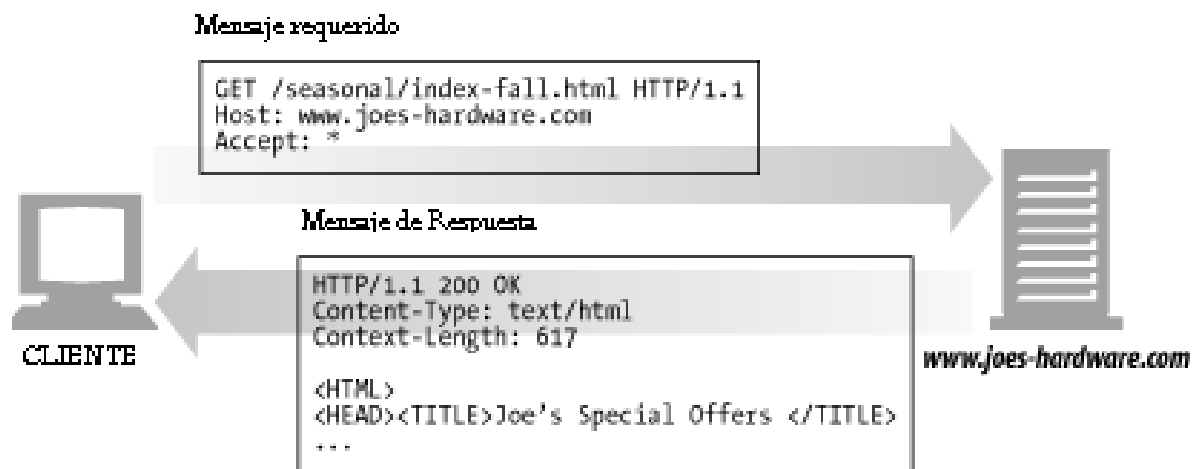
### 2.5.1 Peticiones HTTP

HTTP define una simple interacción petición-respuesta, la cual puede ser vista como una transacción Web. Una petición HTTP incluye varias partes:

- **Método:** el cual especifica la acción legal a realizarse [24] . Los tipos de métodos HTTP son: GET, POST, PUT, DELETE, HEAD, TRACE y OPTIONS. En la figura 12 se puede observar el Método GET en el mensaje de requerimiento, mientras que en la figura 13 se observa el Método PUT [25].

- **URL:** El cual identifica el nombre y la dirección de la información requerida como se observa en las figuras 12 y 13 [25].
- **Información Complementaria:** El tipo de documentos que se puede recibir, autenticación, aspectos de caching, etc [25]. En la Figura 13 se encuentra como información complementaria el tipo de contenido y el tamaño en el mensaje requerido, mientras que en la figura 12 no se registra información complementaria en el mensaje requerido.

Como ejemplo se encuentra uno de los métodos mas populares, el método GET, el cual se usa cuando se da un clic sobre un link o se digita un URL en la barra de un navegador, básicamente es una petición para recibir el contenido de la página Web solicitada como se observa en la figura 12 [25].



**Figura 12:** Ejemplo de Petición HTTP método GET  
Tomada de [24] y modificada por el autor.

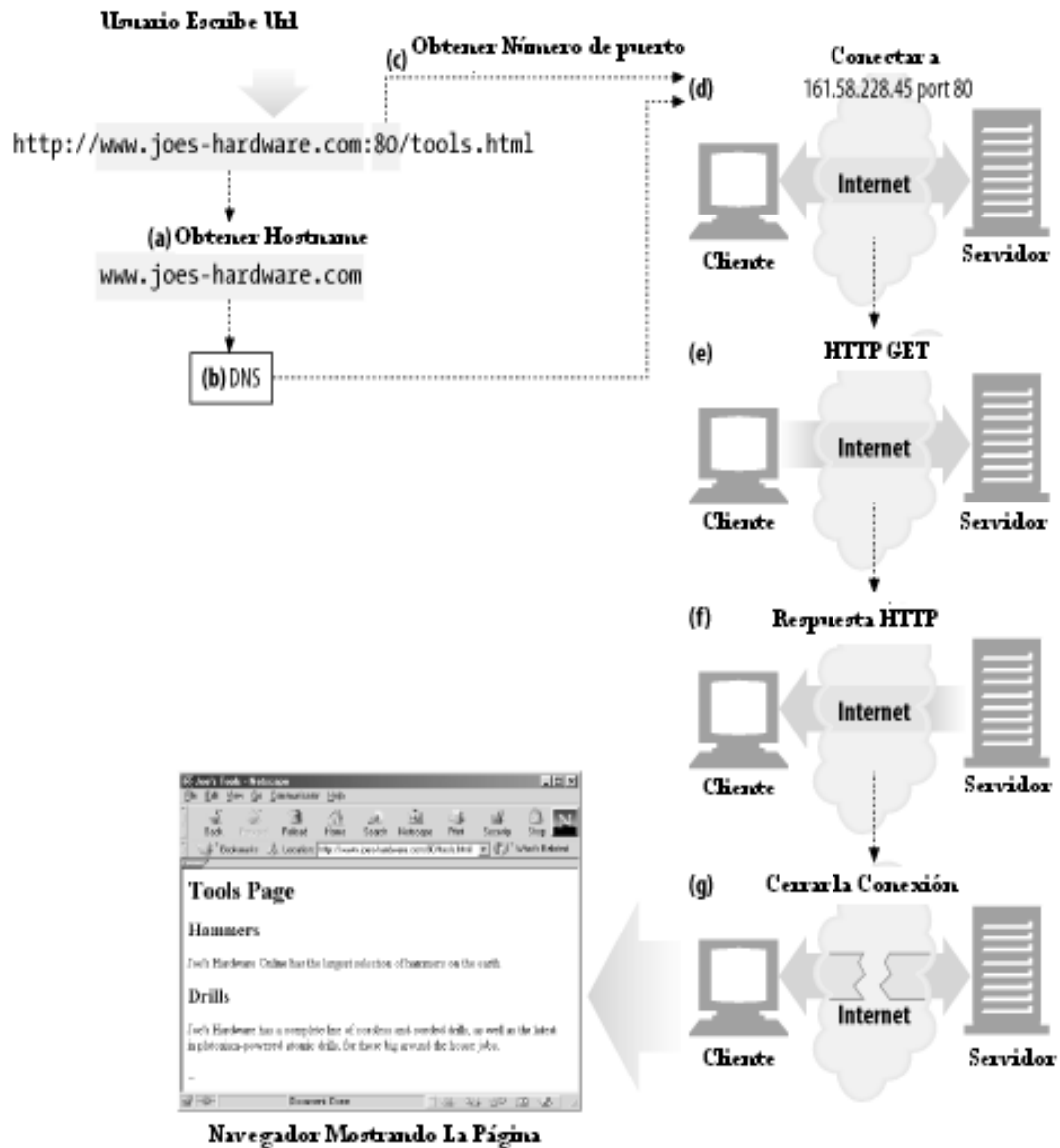
Otro ejemplo de una petición HTTP, usando el Método PUT, el cual permite al enviar contenido el cual será almacenado en el servidor [25]. La respuesta del servidor es que el contenido se ha creado o actualizado satisfactoriamente.



**Figura 13:** Ejemplo de Petición HTTP método PUT  
Tomada de [24] y modificada por el autor.

## 2.5.2 Peticiones HTTP sobre Conexiones TCP

HTTP es considerado un protocolo sin estado debido a que no incluye el concepto de sesión o interacción mas allá de emitir documentos solicitados. Los principales pasos que intervienen en una interacción petición-respuesta en HTTP sobre TCP se observan en la figura 14 [24].



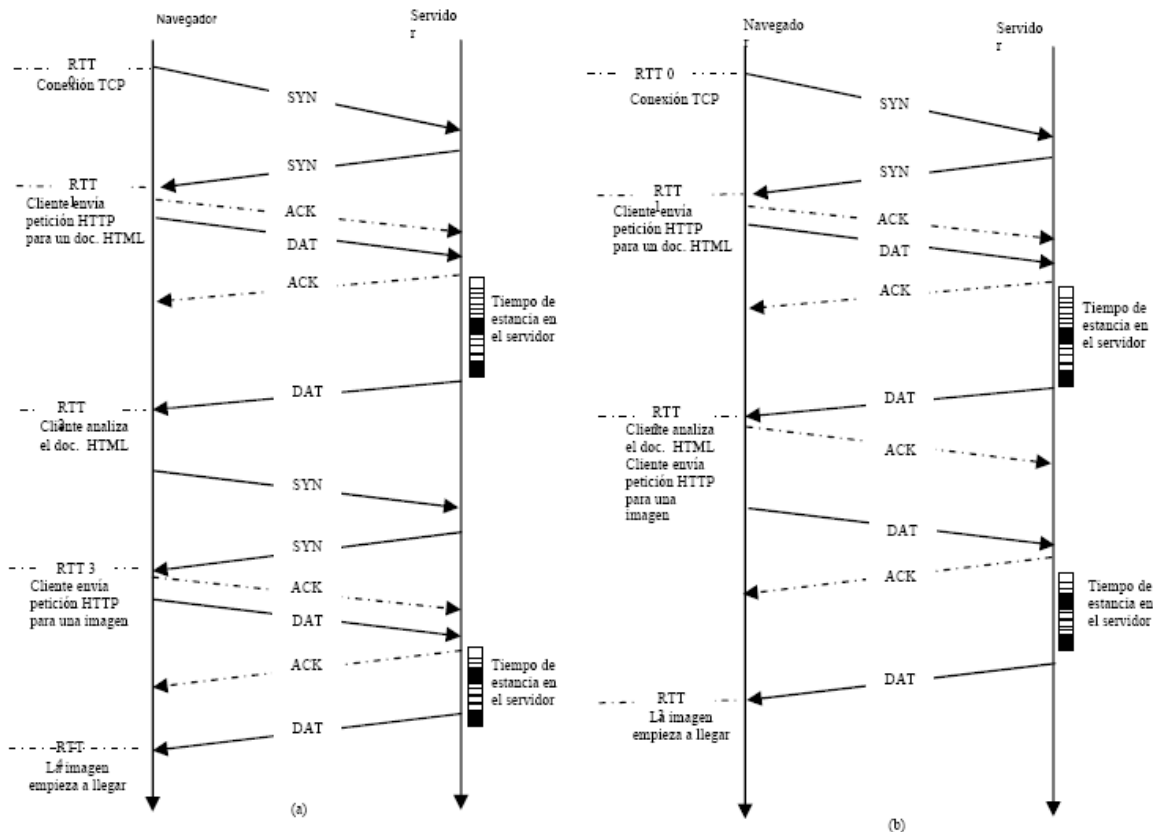
**Figura 14: HTTP sobre TCP**  
Tomada de [24] y modificada por el autor.

En la versión original del protocolo HTTP 1.0 [43], una conversación está restringida a la transferencia de un solo objeto (documento, imagen, etc.) a la vez, lo cual quiere decir que cada una es independiente de la otra como se muestra en la Figura 15 (a). Una página con texto y muchas imágenes pequeñas genera muchas conexiones por separado para el texto y



para cada imagen. Ya que la mayoría de objetos en la Web son pequeños, una fracción alta de paquetes intercambiados entre clientes y servidores son simplemente paquetes de control TCP utilizados en la apertura y cierre de conexiones [42].

Por otra parte, en la versión HTTP 1.1 [43], se usa la persistencia, que permite mantener la misma conexión TCP en todas peticiones HTTP de una misma página. Es decir se usa solo una conexión TCP para trasportar múltiples peticiones HTTP, eliminando el costo de varias aperturas y cierres de conexión como se observa en la figura 15 (b) [42].



**Figura 15:** (a) HTTP 1.0 (b) HTTP 1.1  
Tomada de [42].

Es importante resaltar que en los datos sobre la red de la UPB seccional Bucaramanga tomados en Mayo de 2006 y registrados en [2] solo se evidencia la presencia del protocolo HTTP 1.0.

## **2.6 MODELO DE RED DE COLA CERRADA CQN**

### **2.6.1 Introducción**

Como se menciona en la sección 1.2, este trabajo se enfoca en estimar la capacidad de canal de una red universitaria y surgió como la continuación de los trabajos, “Simulación y análisis de tráfico presente en la red de datos de la universidad Pontificia Bolivariana Seccional Bucaramanga” [2] y “Caracterización y simulación del tráfico de redes LAN mediante el modelo MMPP” [15].

En estos trabajos se realizó la medición sobre el tráfico de la red de la Universidad Pontificia Bolivariana Seccional Bucaramanga [2] y se presentó un método para caracterizar y simular el tráfico de acceso a Internet de cualquier red LAN con comportamiento autosimilar empleando el modelo de cola D-MMPP/D/1 [15].

No obstante, aun era necesaria la obtención de la capacidad de canal en función de la cantidad de usuarios, lo cual permitiría una precisa caracterización del desempeño de la red. Por lo anterior y debido a las características del tráfico señaladas en [2] y [15], (ver sección

3.2 Descripción de la Red y 3.3 Caracterización de las trazas de tráfico), se determino que el modelo de red debería cumplir con las siguientes condiciones:

- Modelar Tráfico Web.
- Modelar el flujo de tráfico a través de un enlace cuello de botella.
- Relacionar la cantidad de usuarios y la capacidad de canal.
- Modelar el comportamiento activo y pasivo de cada usuario.

Para dimensionar la capacidad de canal se tuvieron en cuenta los siguientes modelos:

- El perfil 2Bucket. Este perfil logra el mejor rendimiento de la tasa media a largo plazo de control flujo de tráfico y también permite a la red controlar el tamaño de ráfaga de datos [44]. No obstante no relaciona la cantidad de usuarios y la capacidad de canal.
- El sistema GPS (Procesador Compartido Generalizado) con fuentes del tipo *Markov Modulated Fluid Processes (MMFP)* es un modelo en el que  $N$  clases de tráfico comparten un servidor determinista con tasa de servicio garantizada, el cual es bastante preciso para tráfico de Voz sobre IP y Video (VBR) [45], lo cual no garantiza un ajuste adecuado al tráfico Web.
- EL modelo M/G/1, contiene las características necesarias para modelar las tramas de tráfico multiplexadas, en la hora pico, con una tasa media de servicio, en un canal compartido, tomando en cuenta que el número de fuentes es infinito [46], se tomo este

modelo como punto de partida para encontrar un modelo que estuviera acorde a las características propuestas.

- El modelo de red de cola cerrada, el cual ha sido utilizado exhaustivamente para el análisis y diseño de sistemas y redes de computadoras en [3] y [4], también conocido como el modelo de población finita con un único recurso o servidor en [18] o como “*Palm's Machine Repair Model*” en [23], puede usarse para modelar tráfico elástico debido a que su control realimentado permite modelar las características del control de congestión implementado por TCP, utilizado por el tráfico WEB o HTTP. Tiene un solo servidor o recurso que simula un solo enlace. Relaciona la cantidad de usuarios con la capacidad de canal, teniendo en cuenta el comportamiento activo y pasivo de los usuarios. Por lo anterior, el modelo de red de cola cerrada (CQN) fue el escogido para ser implementado en el presente trabajo.

### **2.6.2 Descripción del fenómeno a modelar**

Una eficaz asignación del ancho de banda es razonable para las clases de tráfico, tales como paquetes de voz (30kbps – 100kbps, por llamada) y paquetes de vídeo (50 Mbps -100 Mbps, dependiendo del tamaño de la imagen) donde es necesaria una asignación eficaz del ancho de banda [47]. Sin embargo, para aplicaciones elásticas de datos que se adaptan a la disponibilidad del ancho de banda, el concepto de un efectivo ancho de banda parece dudoso [3]. Típicas aplicaciones de tráfico elástico son la transferencia de archivos que soporta el correo electrónico o el tráfico Web.

En las aplicaciones de datos elásticas, el usuario y también el diseñador de la red, están preocupados con la demora en la transferencia de un archivo. De este modo, el tiempo en la transferencia de archivos tiene un criterio de desempeño natural para las aplicaciones elásticas (En contraste, para las aplicaciones en tiempo real, como la voz, la demora y la variación en la demora son un tema de interés) [47].

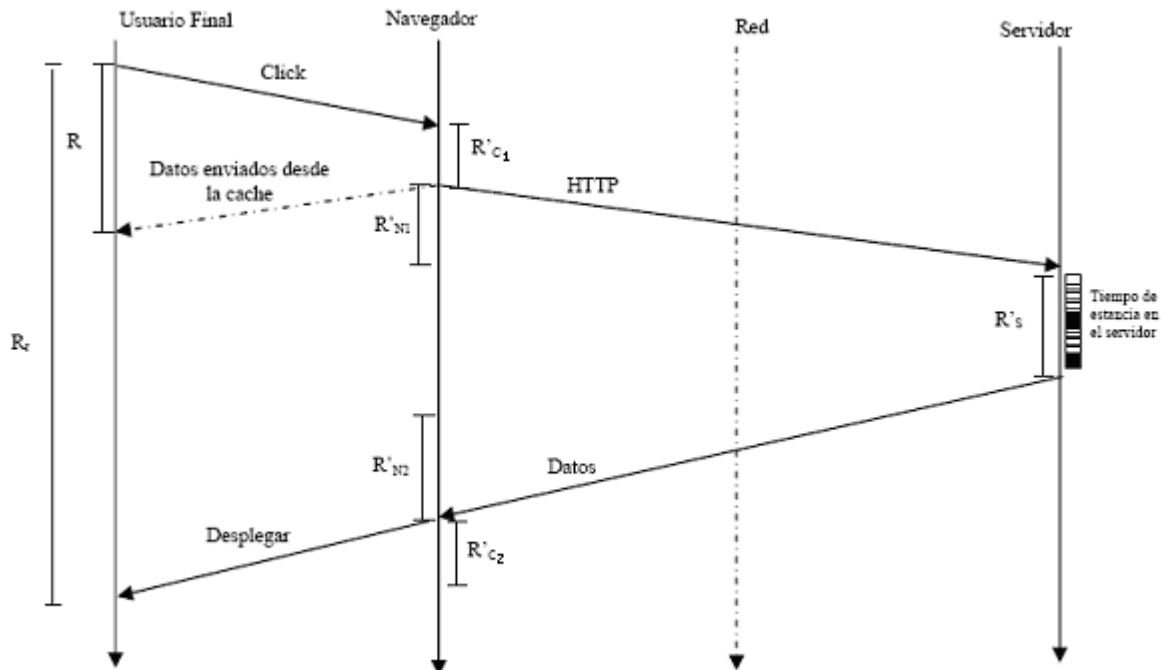
Se puede observar en la figura 16 que se encuentran representados los retardos en una transacción Web. Teniendo en cuenta el caso especial de que la página solicitada no se encuentra en caché, el tiempo de respuesta de una petición es igual a la suma de los tiempos de retardo de la petición en todos los recursos que intervinieron así [42]:

$$\mathbf{R}_t = \mathbf{R}_{\text{Navegador}} + \mathbf{R}_{\text{Red}} + \mathbf{R}_{\text{Servidor}}$$

**Ecuación 10**

Donde:

- $\mathbf{R}_t$  : Tiempo de respuesta total a una petición.
- $\mathbf{R}_{\text{Navegador}}(\mathbf{R}'c1+\mathbf{R}'c2)$  : Tiempo de respuesta del navegador. Es la suma del tiempo de petición y respuesta.
- $\mathbf{R}_{\text{Red}}(\mathbf{R}'_N1+\mathbf{R}'_N2)$ : Tiempo de respuesta de la red. Es la suma del tiempo de petición y respuesta.
- $\mathbf{R}_{\text{Servidor}}(\mathbf{R}'s)$ : Tiempo de respuesta del servidor.



**Figura 16:** Estructura de una transacción WEB  
Tomado de [42].

Dado que el tamaño de los archivos varía enormemente, un simple retardo, como por ejemplo 100ms, no es perceptible para todos los archivos. En cambio, el retardo debería ser normalizado por el tamaño del archivo. Resulta aun más conveniente que consideremos la reciprocidad, de modo que el objetivo es el rendimiento en términos del ancho de banda, en bits/s [42].

### 2.6.3 Formalización del modelo de Red de Cola Cerrada

Por simplicidad, en el presente documento se utilizará el termino “fuente” para referirse al número de generadores de tareas o usuarios. El “enlace” representa el objeto a ser

dimensionado. El tamaño del enlace será calculado para soportar  $N$  fuentes ó usuarios; tomando el punto de vista de que hay un número estático de usuarios presentes, se representará un modelo de población finita con un único recurso representado por un único enlace.

Cada fuente o usuario alterna entre dos fases: ocio (*idle*) o activa (*active*). Tal que, tras generar una tarea durante la fase activa, esta accede al sistema de espera y, una vez que el procesamiento de esta tarea ha terminado, la fuente espera un tiempo aleatorio, que ha sido llamado de ocio, antes de generar la tarea siguiente. Así, el número de tareas del sistema es constante  $K$  y puede denotarse de la siguiente manera [18]:

$$K = O + A$$

**Ecuación 11**

Donde:

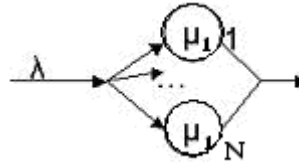
$O \triangleq$  **Número de usuarios en ocio**

$A \triangleq$  **Número de tareas en el sistema de colas**

**Ecuación 12**

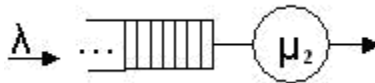
El número fijo de usuarios y el hecho de que alternen entre dos fases hace aceptable el uso de un modelo de red de cola cerrada o CQN [3], [13], [23] con dos tipos de servidores los cuales se encuentran representados así::

- En la figura 18 se puede observar el nodo de servidor infinito (IS) (equivalentemente el número de servidores igual al número de fuentes denotado por  $N$ ) del tipo  $M/G/N$  que modela las fuentes mientras éstas se encuentran en la fase de ocio [3], [13], [23].



**Figura 17:** Modelo  $M/G/N$   
Tomado de [3]

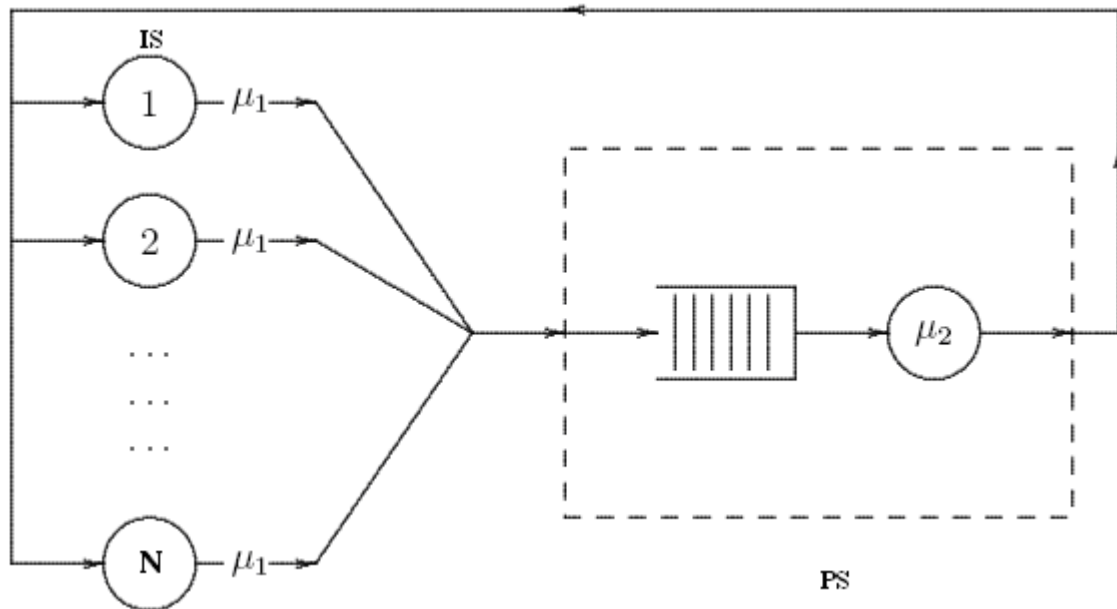
- En la figura 19 se observa el servidor de procesador compartido (PS) del tipo  $M/M/1$ , con la particularidad de que el número de recursos o servidores se reduce a uno, que representa el puerto de salida de la red y el enlace de cuello de botella [3], [13], [23].



**Figura 18:** Modelo  $M/M/1$ .  
Tomado de [3]

Además, como se observa en la figura 19, el modelo CQN es un sistema realimentado, que simula el efecto del algoritmo de control de la ventana de congestión de TCP relacionado en el sección 2.4.3 conocido como arranque lento (slow start) [3].





**Figura 19:** Modelo CQN.  
Tomado de [23]

Como no hay arribo o salida neta del sistema en una red cerrada; la carga de los enlaces  $\lambda$ , es la expresión de la conservación de flujo, igual el flujo de entrada y salida de la cola. En consecuencia (véase figuras 17, 18 y 19), se puede establecer que la tasa media de llegadas (salidas) de tareas al (del) sistema será igual a  $\lambda$ .

#### 2.6.4 Parámetros del Modelo de Red de Cola Cerrada

Para el caso en particular donde el objetivo es la obtención del ancho de banda en un enlace cuello de botella; se pretende obtener una sencilla expresión con el fin de determinar un ancho de banda efectivo para una fuente de tráfico elástico; para lo cual se enuncian los siguientes parámetros:

- El número de Fuentes  $N$ .
- La media de servicio en el nodo **PS** denotada por  $\mu_2^{-1}(s)$ .
- La media de tiempo en el nodo **IS** denotado por  $\mu_1^{-1}(s)$ .
- La media de tamaño de objeto denotado por  $f$  (**bits**).
- La capacidad del enlace seleccionado denotada por  $B$  (**bps**).
- La capacidad de enlace de una fuente denotado por  $b$  (**bps**).

La media de tiempo de servicio en el nodo PS representa la media del tiempo para transmitir un archivo en el enlace seleccionado, sin tener en cuenta otros archivos presentes [3].Luego,

$$\mu_2^{-1} = f / B.$$

**Ecuación 13**

La media de tiempo en el nodo IS representa la media de tiempo entre el inicio de la transferencia de un archivo por una fuente [3].

El único aspecto no estándar del modelo CQN es la entidad que representa la "tarea". Una tarea en el modelo CQN representa un objeto<sup>1</sup> (o varios objetos) y no se encuentra al nivel de paquetes porque introduciría un nivel de detalle y complejidad que no es necesario para el desarrollo de este trabajo [3] .

---

<sup>1</sup> EL modelo CQN representa un archivo, para el tráfico Web de la UPB con HTTP 1.0 (vease sección 2.5.2), representa objetos.

### 2.6.5 Ancho de Banda Efectivo

$B_s$  denota el ancho de banda que una fuente arbitraria obtiene en un estado estable. Este ancho de banda por fuente se define como:

$$B_s = B / Q'_1$$

**Ecuación 14**

Donde:

- B representa el ancho de banda del enlace
- $Q'_1$  es el número de tareas en el nodo PS, (teniendo en cuenta que este nodo no está vacío).

Además, considerando el criterio de funcionamiento sobre la media y sobre la probabilidad de cola de  $B_s$  según [3].

- Por definición la probabilidad de que  $Q'_1$  sea igual a un valor “n” teniendo  $n = 1, \dots, N$  es igual a:

$$\Pr(Q'_1 = n) = \frac{\Pr(Q_1 = n)}{\Pr(Q_1 > 0)}, n = 1, 2, \dots, N.$$

**Ecuación 15**

- Se tiene que el Valor esperado de  $B_s$  será mayor o igual al valor escogido dentro del rango de  $10^4$ - $10^6$  bit/s que representa la capacidad de canal garantizado a cada usuario (**b**) [3]:

$$E[B_s] \geq b$$

**Ecuación 16**

- Además hay un interés principal en un escenario donde la red de datos este congestionada dado que durante ese tiempo los recursos de la red tienden a limitar el factor de velocidad obtenido por la conexiones de tráfico elástico [3]. En consecuencia el factor de utilización en el enlace a dimensionar deberá ser exponencialmente cercano o mayor a 1. Es decir, para considerar las aproximaciones de tráfico Web con características autosimilares<sup>2</sup> [40], se tendrá en cuenta solo la región donde  $N$  es grande y  $\Pr(Q'_1 > 0)$  tienda exponencialmente a 1, así:

$$\rho = N \frac{\mu_1}{\mu_2} > 1$$

**Ecuación 17**

En consecuencia, el problema de dimensionamiento se reduce a minimizar  $\mathbf{B}$  de manera que los criterios descritos anteriormente estén satisfechos.

### **2.6.6 El criterio del valor medio**

El algoritmo de valor medio (MVA, *mean value algorithm*) es un mecanismo para medir el rendimiento de una redes de cola. Combina de excelente manera dos resultados principales de la teoría de puesta en fila de espera: el teorema de llegada y la ley de Little (ver sección 2.3). El algoritmo fue publicado por primera vez por Lavenberg y Reiser en 1980 [23].

---

<sup>2</sup> Autosimilar: fenómeno donde una característica particular es similar en diferentes escalas de espacio o tiempo. [40]

En el mismo se considera una red de puesta en fila con  $K$  nodos y  $S$  clientes (todos pertenecientes a una sola cadena). Las cargas relativas de los nodos se simbolizan por [23]:

$$\alpha_k \quad (k = 1, 2, \dots, K)$$

**Ecuación 18**

El algoritmo es recurrente en el número de clientes, es decir, una red con  $x$  clientes se evalúa a partir de una red con  $x-1$  clientes. Supóngase que el promedio de la cantidad de clientes en el nodo  $k$  es  $L_k(x)$  donde  $x$  es el número de clientes total en la red. Obviamente [23]:

$$\sum_{k=1}^K L_k(x) = x$$

**Ecuación 19**

El algoritmo actúa recurrentemente en dos pasos:

**Paso 1:** Aumentar el número de clientes de  $x$  a  $(x + 1)$ . Conforme al teorema de llegada, el  $(x + 1)$ -ésimo cliente verá el sistema como si éste tuviera  $x$  clientes en equilibrio estadístico. Por tanto, el promedio de tiempo de ocupación (tiempo de espera, + tiempo en servicio) en el nodo  $k$  es [23]:

- Para M/M/1, M/G/1-PS, M/G/1, y LCFS-PR:

$$W_k(x+1) = \{L_k(x) + 1\} \cdot s_k$$

**Ecuación 20**

- Para M/G/∞:

$$W_k(x+1) = s_k$$

**Ecuación 21**

Donde:

- $s_k$  es el promedio del tiempo de servicio en el nodo  $k$  que tiene  $n_k$  servidores.

En razón que solo se calcula el tiempo medio de espera, se puede suponer el criterio de fila de espera *FCFS* (First Come First Served) descrito en la sección 2.3.4 .

**Paso 2:** Se aplica la ley de Little ( $L = \lambda \cdot W$ ) (ver sección 2.3), que es válida para todos los sistemas en equilibrio estadístico. Para el nodo  $k$  se tiene [23]:

$$L_k = \lambda_k \cdot W_k,$$

**Ecuación 22**

Donde:

- $\lambda_k$  es la velocidad de llegada relativa al nodo  $k$ .

Aplicando el teorema de llegada a la ecuación 21, se tiene:

$$L(x+1) = c \cdot \lambda_k \cdot W_k(x+1)$$

**Ecuación 23**

Donde:

- La constante  $c$  se obtiene del número total de clientes:

$$\sum_{k=1}^K L_k(x+1) = x+1$$

**Ecuación 24**

Mediante estos dos pasos se ha efectuado la recursión<sup>3</sup> de  $x$  a  $(x + 1)$  clientes. Para  $x = 1$  no habrá tiempo de espera en el sistema y  $W_k(1)$  es igual al promedio del tiempo de servicio  $s_k$ .

En una red de cola cerrada con  $N$  fuentes y dos nodos, si se aplica el algoritmo se tiene que velocidades de visita relativa son idénticas, pues un cliente visita alternativamente el nodo 1 y el nodo 2 es igual a  $\lambda_1 = \lambda_2 = 1$ . Por tanto se tiene que:

	<b>Nodo 1 (IS)</b>	<b>Nodo 2 (PS)</b>
$S = 1$	$W_1(1) = \frac{\mu_2}{\mu_1}$	$W_2(1) = \frac{\mu_2}{\mu_1}$
	$L_1(1) = c \cdot 1 \cdot \frac{\mu_2}{\mu_1}$	$L_2(1) = c \cdot 1 \cdot 1$
	$L_1(1) = \frac{\frac{\mu_2}{\mu_1}}{1 + \frac{\mu_2}{\mu_1}}$	$L_2(1) = \frac{1}{1 + \frac{\mu_2}{\mu_1}}$

**Tabla 2:** Aplicación del MVA sobre el modelo de red de cola cerrada donde  $S=1$ . Tomado de [23], modificado por el autor.

<sup>3</sup> Un objeto es recursivo si en su definición se nombra a sí mismo [23]

	Nodo 1 (IS)	Nodo 2 (PS)
S = 2	$W_1(2) = \frac{\mu_2}{\mu_1}$ $L_1(2) = c \cdot 1 \cdot \frac{\mu_2}{\mu_1}$ $L_1(2) = \frac{\mu_2}{\mu_1} \cdot \frac{1 + \frac{\mu_2}{\mu_1}}{1 + \frac{\mu_2}{\mu_1} + \frac{\left(\frac{\mu_2}{\mu_1}\right)^2}{2!}}$	$W_2(1) = 1 + \frac{1}{1 + \frac{\mu_2}{\mu_1}}$ $L_2(2) = c \cdot 1 \cdot \left(1 + \frac{1}{1 + \frac{\mu_2}{\mu_1}}\right)$ $L_2(2) = 2 - \frac{1 + \frac{\mu_2}{\mu_1}}{1 + \frac{\mu_2}{\mu_1} + \frac{\left(\frac{\mu_2}{\mu_1}\right)^2}{2!}}$

**Tabla 3:** Aplicación del MVA sobre el modelo de red de cola cerrada donde S=2.  
Tomado de [23], modificado por el autor.

Obteniendo como formula general:

	Nodo 1	Nodo 2
S = x	$W_1(x) = \frac{\mu_2}{\mu_1}$ $L_1(x) = c \cdot \frac{\mu_2}{\mu_1}$ $L_1(x) = \frac{\mu_2}{\mu_1} \cdot \left\{1 - E_x\left(\frac{\mu_2}{\mu_1}\right)\right\}$	$W_1(x) = 1 + L_2(x-1)$ $L_1(x) = 1 + \{1 + L_2(x-1)\}$ $L_1(x) = x - \frac{\mu_2}{\mu_1} \cdot \left\{1 - E_x\left(\frac{\mu_2}{\mu_1}\right)\right\}$

**Tabla 4:** Aplicación del MVA sobre el modelo de red de cola cerrada donde S=x.



Tomado de [23], modificado por el autor.

De esto se obtiene la constante de normalización  $c = 1 - E_x(\mu_1/\mu_2)$  y se calcula para el  $(x+1)$ -ésimo cliente:

$$L_1(x+1) + L_2(x+1) = c \cdot \frac{\mu_1}{\mu_2} + c \cdot \{1 + L_2(x)\}$$

$$L_1(x+1) + L_2(x+1) = c \cdot \frac{\mu_1}{\mu_2} + c \cdot \left\{1 + x - \frac{\mu_1}{\mu_2} \cdot (1 - E_x)\right\}$$

$$L_1(x+1) + L_2(x+1) = x + 1$$

$$E_{x+1} = \frac{\frac{\mu_1}{\mu_2} \cdot E_x}{x + 1 + \frac{\mu_1}{\mu_2} \cdot E_x}$$

**Ecuación 25**

Se sabe que  $c = 1 - E_{x+1}$ . Por lo tanto la ecuación 25, es la formula de recursión para la fórmula B de Erlang [23]. Un ejemplo de la Formula B de Erlang se ilustra en la figura 20.

$$\begin{aligned} E_0(2) &= 1, \\ E_1(2) &= \frac{2 \cdot 1}{1 + 2 \cdot 1} = \frac{2}{3}, \\ E_2(2) &= \frac{2 \cdot \frac{2}{3}}{2 + 2 \cdot \frac{2}{3}} = \frac{2}{5}, \\ E_3(2) &= \frac{2 \cdot \frac{2}{5}}{3 + 2 \cdot \frac{2}{5}} = \frac{4}{19}, \\ E_4(2) &= \frac{2 \cdot \frac{4}{19}}{4 + 2 \cdot \frac{4}{19}} = \frac{2}{21}, \\ E_5(2) &= \frac{2 \cdot \frac{2}{21}}{5 + 2 \cdot \frac{2}{21}} = \frac{4}{109}, \\ E_6(2) &= \frac{2 \cdot \frac{4}{109}}{6 + 2 \cdot \frac{4}{109}} = \frac{4}{331} = 0.0121. \end{aligned}$$

**Figura 20:** Ejemplo de la Formula B de Erlang.

Tomado de [23]

De lo anterior se obtiene la aproximación asintótica del número medio de tareas en el sistema de espera, derivada del modelo de Erlang [3], [13].

$$Q_1 = K - \left(1 - \frac{r}{1+r}\right) \cdot r,$$

**Ecuación 26**

Donde:

- $r = \frac{\mu_1}{\mu_2}$
- K se encuentra enunciado en la ecuación 11.

Como ejemplo se tiene que si  $K=1$ , entonces  $Q_1$  es igual a  $\frac{1}{1+r}$ , tal como se observa en la tabla 2.

### **2.6.7 Solución aproximada por el desarrollo del criterio de media**

Usando el Criterio de Media (véase ecuación 26), el criterio propuesto para el Ancho de Banda Efectivo (véase ecuación 16) y teniendo en cuenta el factor de utilización del enlace (véase ecuación 17) según [3] se obtiene la siguiente expresión:

$$\mathbf{B} = \mathbf{N} * \mathbf{h}$$

**Ecuación 27**

Donde se obtiene una simple aproximación para satisfacer el criterio de ancho de banda efectivo para cada fuente con una velocidad  $h$  [3]. Donde  $h$  es la media armónica de dos velocidades que se encuentran naturalmente asociadas al tráfico de datos elástico dentro de los límites (impuesto por el administrador de la red y denotado por  $b$ ) y condiciones de la red (determinado por el producto la tasa media de transferencia del nodo PS y el tamaño medio de los objetos transferidos).

$$h = \left( \frac{1}{b} + \frac{1}{\mu_1 f} \right)^{-1}$$

**Ecuación 28**

Suponiendo que las fuentes tienen mucho que enviar y tienen un ancho de banda  $B$  en el enlace, debe existir una restricción sobre la velocidad de transmisión, dado que seguramente todos los usuarios tienen una tarea en el nodo PS, teniendo como consecuencia que  $Q_1$  es igual a  $N$ , por tanto  $B_s = B/N$  donde cada usuario recibe un ancho de banda igual a  $b$ . En el otro extremo, el enlace a ser dimensionado, no tiene mucho tráfico y la red no opone ninguna restricción, entonces cada usuario puede transmitir a una velocidad promedio de  $\mu_1 * f$ .

De lo anterior se ha obtenido una expresión simple y robusta para dimensionar el ancho de banda para tráfico de datos elástico. La derivación esta basada en una aproximación

asintótica, para un Modelo de Red de Cola Cerrada con tráfico autosimilar [3], donde la distribución de las variables aleatorias es pertinente solo utilizando la media, con un interés particular en la media de los tamaños de los archivos y no sus características de tráfico autosimilar.

Es importante resaltar que la expresión de ancho de banda efectivo con el criterio de media para tráfico elástico es fácilmente invertible para obtener el número de usuarios que puede soportar una red dado un valor de ancho de banda.

## **2.7 HERRAMIENTAS DE SOFTWARE UTILIZADAS**

Para el desarrollo del presente trabajo se hizo necesario el uso de distintos tipos de software, para capturar y caracterizar las tramas de tráfico de acuerdo a los parámetros del modelo de red de cola cerrada. Este software es de licencia libre y se encuentra disponible en la Web, con el fin de que la metodología propuesta pueda emularse satisfactoriamente.

### **2.7.1 Wireshark**

Los administradores de sistemas, ingenieros de redes, ingenieros de seguridad, los operadores del sistema y programadores utilizan analizadores de red. El análisis de redes

es el proceso de captura de tráfico de la red para determinar lo que está sucediendo en la misma.

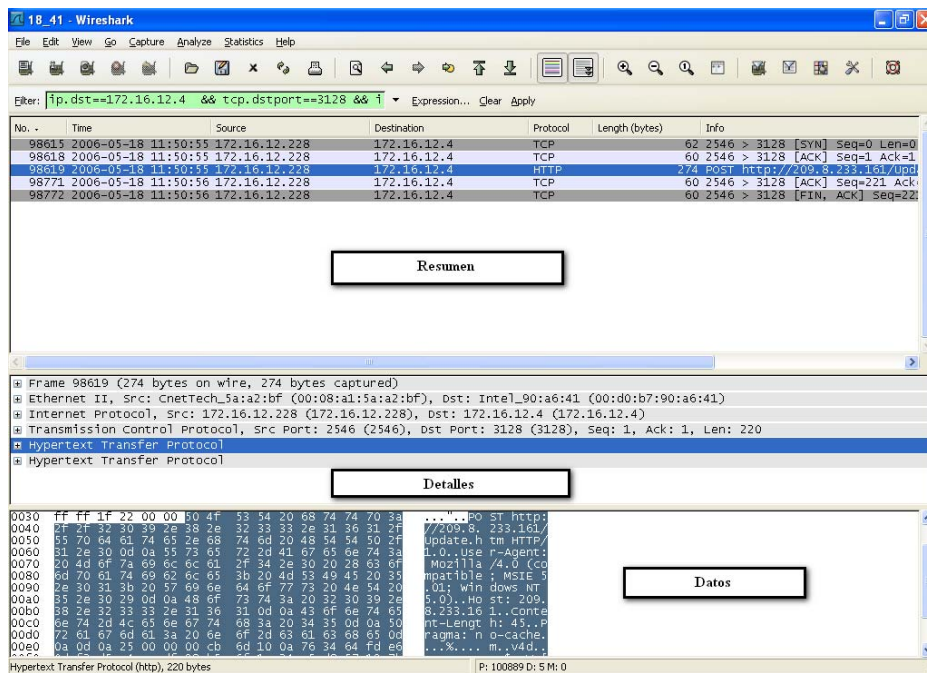
Una analizador red decodifica los paquetes de datos de los protocolos comunes y muestra el tráfico de la red en formato legible, mientras que un sniffer es un programa que monitorea los datos que viajan a través de una red de manera no autorizada; estos últimos son peligrosos para la seguridad de la red porque son difíciles de detectar y se pueden insertar en casi cualquier lugar, lo que los convierte en uno de los armas favoritas de los hackers.

Un analizador de red puede ser un dispositivo de hardware con software especializado, o simplemente un software que se instala en un computador de escritorio o portátil. Las diferencias entre los analizadores de red dependen de sus funciones, tales como el número de protocolos que puede decodificar, la interfaz de usuario, su representación gráfica y la capacidad estadística. Otras diferencias incluyen la capacidad de inferencia (por ejemplo, análisis de expertos) y la calidad de los paquetes decodifica. Aunque varios analizadores de red decodifican los mismos protocolos, algunos funcionan mejor que otros de su entorno.

Wireshark es uno de los mejores analizadores de red disponibles; desarrollado como software libre con calidad de software comercial. Cuenta con numerosas funciones, una agradable interfaz gráfica de usuario (GUI, la Figura 21 muestra la pantalla de Wireshark Network Analyzer), más de 400 protocolos decodificados y esta siendo activamente desarrollado y mantenido. Funciona en sistemas basados en UNIX, Mac OS X y Windows. El paquete, está disponible en [www.wireshark.org](http://www.wireshark.org).

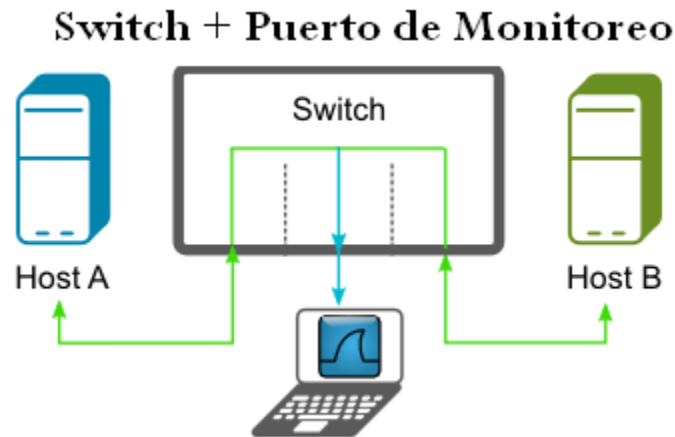
Un típico analizador de red muestra el tráfico capturado en tres paneles según [28] y como puede observarse en la figura 21:

- **Resumen:** Este panel se muestra un resumen de una línea de la captura. Campos que incluyen la fecha, hora, dirección de origen, dirección de destino, y el nombre e información sobre el protocolo de capa más alta.
- **Detalles:** Este panel ofrece todos los detalles (en un árbol-como la estructura) para cada una de las capas contenidas en el interior de la captura de paquetes.
- **Datos:** muestra los datos capturados en bruto en hexadecimal y formato de texto.



**Figura 21:** GUI Wireshark.  
Creada por el autor.

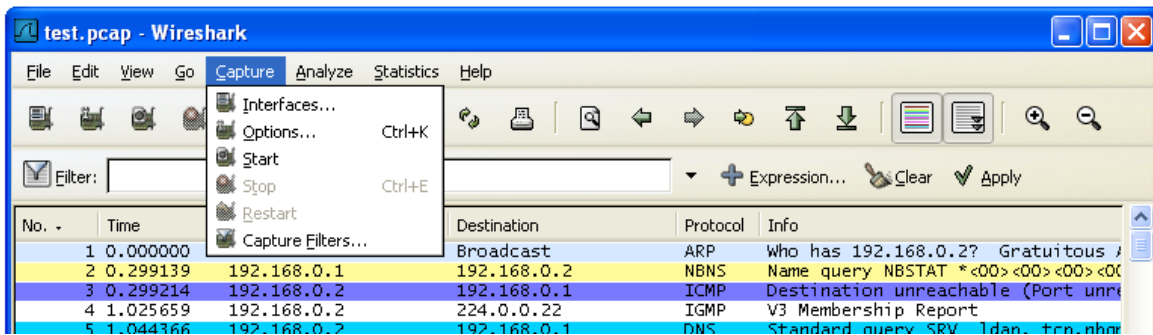
Para capturar el tráfico de la red, se debe tener en cuenta la siguiente topología propuesta en la figura 22:



**Figura 22:** Topología para la Captura de tráfico.  
Tomado de [29], modificado por el autor.

El puerto de Monitoreo del Switch, a veces llamado "puerto espejo", debe utilizarse para poder monitorear todo el tráfico que circula a través del Switch, de lo contrario, solo podrá observarse el tráfico de un puerto en particular. La desventaja de este tipo de capturas es que en el caso de alto tráfico puede presentarse una pérdida de paquetes.

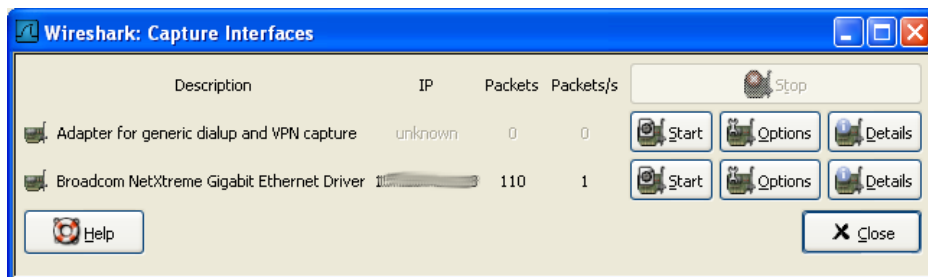
Dentro de la Interfaz de Usuario del analizador de protocolos encontramos el menú captura, mediante el cual se definen los diferentes parámetros de la captura de datos, como se observa en la figura 23:



**Figura 23:** Menú Capture.

Tomado de [29], modificado por el autor.

Al seleccionar la opción *Interfaces*, aparece un cuadro de dialogo, donde se muestran las interfaces o tarjetas de red, con una breve descripción y sus principales características, además, se puede iniciar o detener una captura sobre cualquiera de las interfaces que el software detecte como se observa en la figura 24.



**Figura 24:** Menú Capture Interfaces.

Tomado de [29], modificado por el autor.

No obstante, para una captura mas especializada, es recomendable seleccionar en el menú capture *Options*, ya que en este cuadro de dialogo se puede determinar: la interfaz, el filtro de protocolo ó IP, el máximo tamaño de archivo de captura, el tiempo de captura, entre otras opciones más complejas, como se puede observar en la figura 25.





**Figura 25:** Menú Capture Options.

Tomado de [29], modificado por el autor.

Aunque previamente se han señalado las incontables ventajas del uso del analizador de protocolos Wireshark, es importante resaltar que una de sus principales desventajas es el espacio en el disco que puede ocupar un análisis de red muy largo, debido a que la captura incluye todo lo que circula por el canal, creado incluso la posibilidad de reconstruir los archivos que fueron solicitados por los usuarios de la red.

En consecuencia, un computador de gama media no podrá realizar el análisis de la captura del tráfico de una red en la hora pico, ya que este archivo podría llegar a tener un tamaño en (Mb o Gb) demasiado grande, sobrepasando los recursos del procesador y memoria del computador.

## 2.7.2 Honeysnap

Con la intención de obtener una muestra del tráfico manejable, que permitiera el análisis de los parámetros necesarios para evaluar el modelo de red de cola cerrada, se hizo necesaria la búsqueda de otro software que permitiera analizar de manera simultanea, la gran cantidad de archivos que puede generar la captura de tráfico de una red LAN en la hora pico. Como resultado de esta búsqueda se obtuvo el software de licencia libre HoneySnap.

Honeysnap es una herramienta de línea de comandos diseñada para analizar uno o múltiples archivos de datos pcap y elaborar un análisis que identifique los acontecimientos importantes en el procesado de datos. Otras funciones incluidas son:

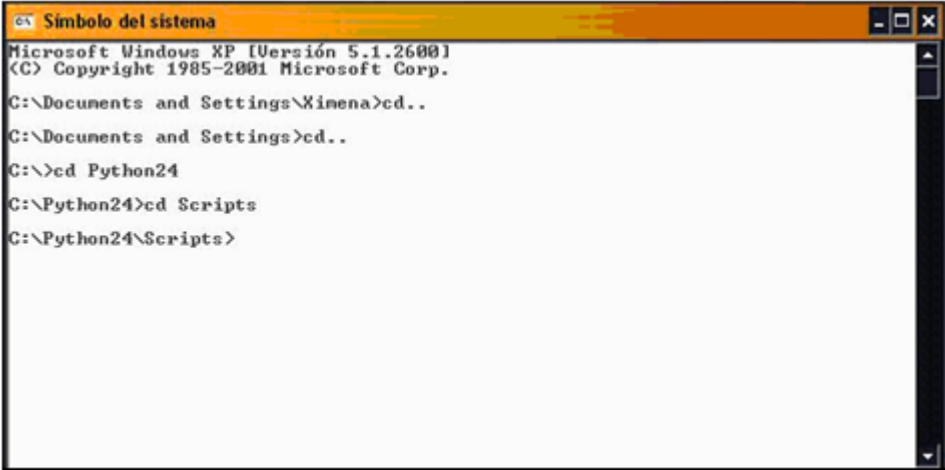
- Visión de conjunto de las conexiones y flujo de paquetes
- Extracción de flujo de las comunicaciones basadas en ASCII.
- Decodificar los protocolos mas comunes de comunicación de Internet.
- Extracción de la transferencia de archivos binarios.
- El resumen de entrada y de salida de las conexiones.

Cada distribución de Honeysnap viene con la documentación sobre cómo instalar, configurar y utilizar Honeysnap. La cual puede obtenerse en la pagina Web <https://projects.honeynet.org/honeysnap/>.

La forma más fácil de empezar a usar Honeysnap es tomar el archivo de muestra honeynet.cfg, modificar la dirección IP objeto de análisis (también llamada por sus creadores “honeypot”), que se encuentre dentro del rango de la red a analizar y que se clave para determinar el comportamiento del tráfico, por ejemplo la dirección IP del Proxy de la red.

A continuación, se seleccionan las opciones de análisis requerido (simplemente se digita la palabra *Yes* enfrente la opción), como el resumen de la información del archivo pcap y las conexiones y entrada y salida al Honeypot o Honeypots.

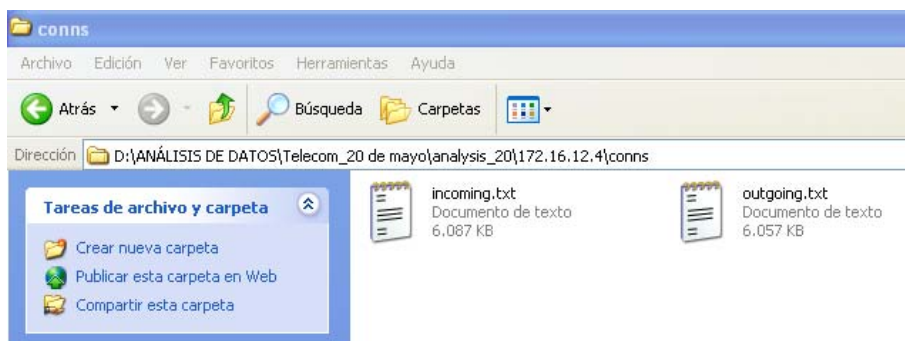
Con la mayoría de las opciones seleccionadas y teniendo en cuenta que Honeysnap es una aplicación modular del lenguaje Python [31], a través de la línea de comando se ingresa al directorio Python24 y posteriormente al directorio Scripts como, se visualiza en la figura 26.



```
Simbolo del sistema
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Ximena>cd..
C:\Documents and Settings>cd..
C:\>cd Python24
C:\Python24>cd Scripts
C:\Python24\Scripts>
```

**Figura 26:** Ingreso a Python en la Línea de Comandos.  
Creado por el autor.

Para finalizar se digita: *honeysnap-c honeynet.cfg file.pcap*, en la línea de comandos, mediante el cual se realizan los análisis requeridos, sobre las IP establecidas como Honeypots y se presentan al usuario mediante archivos de texto. En la figura 26 de Observa, el tipo de archivo generado y su respectivo tamaño, representando la conexiones TCP y UDP, de entrada y salida al honeypot.



**Figura 27:** Archivos generados por Honeysnap.  
Creado por el autor.

En estos archivos solo se muestran datos como:

- La fecha y hora de inicio de transmisión del objeto.
- La fecha y hora de fin de transmisión del objeto.
- La IP origen con su respectivo puerto.
- La IP destino con su respectivo puerto.
- El número de paquetes necesarios para cada conexión.
- La cantidad de bytes que corresponden al tamaño del objeto transmitido.

Eliminando de esta manera, los datos innecesarios contenidos en los archivos pcap de la captura de tráfico, como se observa en las figuras 28 y 29.

Incoming connections for 172.16.12.4

TCP TRAFFIC SUMMARY:

Start	End	Source	Sport	Dest	Dport	Count	Bytes
Fri May 19 20:27:18 2006	Fri May 19 20:29:29 2006	172.16.12.241	1834	172.16.12.4	3128	83	17433
Sat May 20 03:07:43 2006	Sat May 20 03:07:43 2006	172.16.6.5	1086	172.16.12.4	53427	1	0
Sat May 20 00:52:43 2006	Sat May 20 00:52:43 2006	172.16.6.5	1086	172.16.12.4	51648	1	0
Sat May 20 00:02:43 2006	Sat May 20 00:02:43 2006	172.16.6.5	1086	172.16.12.4	50989	1	0
Sat May 20 03:13:43 2006	Sat May 20 03:13:43 2006	172.16.6.5	1086	172.16.12.4	53506	1	0
Fri May 19 21:08:43 2006	Fri May 19 21:08:43 2006	172.16.6.5	1086	172.16.12.4	48630	1	0
Fri May 19 22:05:43 2006	Fri May 19 22:05:43 2006	172.16.6.5	1086	172.16.12.4	49384	1	0
Sat May 20 01:31:43 2006	Sat May 20 01:31:43 2006	172.16.6.5	1086	172.16.12.4	52162	1	0
Sat May 20 00:41:43 2006	Sat May 20 00:41:43 2006	172.16.6.5	1086	172.16.12.4	51503	1	0
Sat May 20 04:27:43 2006	Sat May 20 04:27:43 2006	172.16.6.5	1086	172.16.12.4	54490	1	0
Fri May 19 20:47:43 2006	Fri May 19 20:47:43 2006	172.16.6.5	1086	172.16.12.4	48286	1	0
Sat May 20 05:18:43 2006	Sat May 20 05:18:43 2006	172.16.6.5	1086	172.16.12.4	55153	1	0
Fri May 19 20:26:25 2006	Fri May 19 20:27:17 2006	172.16.12.241	1830	172.16.12.4	3128	10	1610

**Figura 28:** Resumen de Conexiones TCP archivo incoming.txt.  
Creado por el autor.

UDP TRAFFIC SUMMARY:

Start	End	Source	Sport	Dest	Dport	Count	Bytes
Sat May 20 06:34:36 2006	Sat May 20 06:34:48 2006	172.16.12.4	47887	172.16.0.131	53	5	179
Sat May 20 04:04:33 2006	Sat May 20 04:04:33 2006	172.16.12.4	47846	172.16.0.131	53	1	45
Sat May 20 04:04:34 2006	Sat May 20 06:32:57 2006	172.16.12.4	47882	172.16.0.131	53	92	3609
Sat May 20 04:04:33 2006	Sat May 20 04:04:33 2006	172.16.12.4	47845	172.16.0.131	53	4	175
Sat May 20 04:04:34 2006	Sat May 20 04:04:34 2006	172.16.12.4	47881	172.16.0.131	53	1	42
Sat May 20 04:04:33 2006	Sat May 20 04:04:33 2006	172.16.12.4	47840	172.16.0.131	53	1	43
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47739	172.16.0.131	53	1	45
Sat May 20 04:04:27 2006	Sat May 20 04:04:27 2006	172.16.12.4	47775	172.16.0.131	53	5	220
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47734	172.16.0.131	53	1	46
Sat May 20 04:04:26 2006	Sat May 20 04:04:26 2006	172.16.12.4	47770	172.16.0.131	53	6	263
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47733	172.16.0.131	53	1	43
Sat May 20 04:04:26 2006	Sat May 20 04:04:26 2006	172.16.12.4	47769	172.16.0.131	53	1	45
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47728	172.16.0.131	53	1	43
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47764	172.16.0.131	53	1	44
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47763	172.16.0.131	53	1	43
Sat May 20 02:40:40 2006	Sat May 20 03:39:28 2006	172.16.12.4	47722	172.16.0.131	53	1291	59212

**Figura 29:** Resumen de Conexiones UDP archivo outgoing.txt.  
Creado por el autor.

### 2.7.3 Sed

Teniendo el tamaño de los archivos transmitidos y las conexiones TCP y UDP en archivos de texto plano de tamaño manejable; aun es necesario editar estos archivos para eliminar las conexiones UDP (Previamente se determino que este tipo de tráfico es casi nulo[2]), desde un punto de vista inicial sería factible realizarlo de manera manual, teniendo como inconvenientes, los errores generados y el tiempo empleado.

Por lo anterior fue necesario utilizar un editor de contenidos, de licencia libre, llamado Sed; este editor tiene la característica de utilizarse a través de líneas de comandos, con

instrucciones de funcionamiento simples. Las cuales pueden encontrarse en la página Web <http://www.gnu.org/software/sed/manual/sed.html>.

Para usar *Sed*, normalmente se invoca desde la línea de comandos mediante la instrucción:

- ***-InputFile sed,***

Cuyo formato completo sería el siguiente:

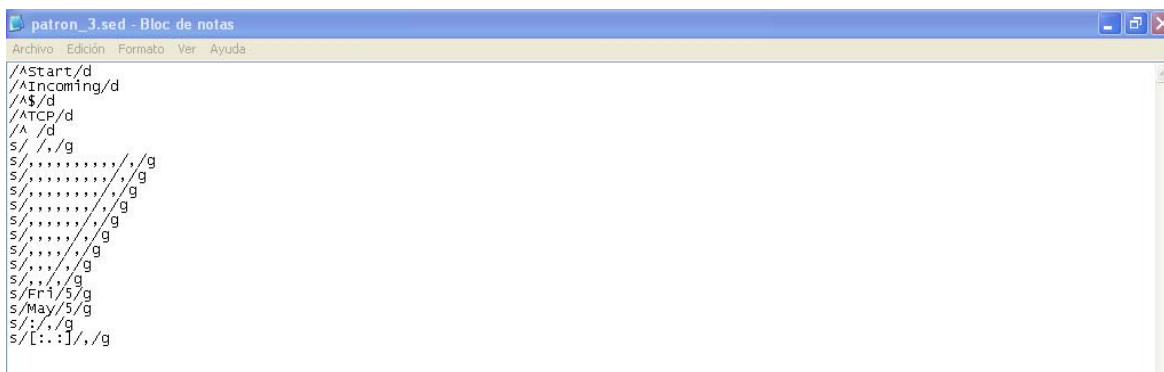
- **OPCIONES sed ... [Guión] [InputFile ...],**

Por otra parte el comando:

- **“-f script-file”,**

Añade los comandos contenidos en el “script-file” al conjunto de comandos que se ejecuta en el procesamiento de la entrada.

Mediante el editor de contenidos Sed, es posible, eliminar líneas que empiecen por determinada palabra o carácter, un ejemplo de esto se observa en las cuatro primeras líneas del archivo `patron_3.sed`; también es posible eliminar líneas en blanco (véase línea 5 de la figura 30), reemplazar caracteres o palabras (véase líneas 6-19 de la figura 30), sobre un archivo de texto de manera rápida y efectiva.



```
patron_3.sed - Bloc de notas
Archivo Edición Formato Ver Ayuda
//start/d
//incoming/d
//$/d
//TCP/d
//A /d
s/ /, /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/..... /g
s/r1/s/g
s/May/s/g
s:/ /g
s/[:.:]/ /g
```

**Figura 30:** Archivo Patron\_3.sed .  
Creado por el autor.

El comando “– Ayuda”, Imprime un mensaje de uso de una breve exposición de las opciones de línea de comandos, la dirección de presentación de informes de errores y a continuación, salir.

#### **2.7.4 Network Simulator 2 (NS-2)**

En la actualidad existen varios lenguajes y paquetes para simular redes de computadoras y en general todas estas herramientas ayudan a modelar entornos de red para ser analizados. Algunos utilizan scripts, otros cuentan con entornos gráficos, sin embargo tienen el inconveniente de ser comerciales y no ofrecen su código libre para investigación y extensión. NS-2 es una herramienta fuertemente utilizada en la investigación, ofrece documentación de su código y existe disponibilidad por parte de sus desarrolladores para apoyar proyectos con NS.

Network Simulator también llamado NS, pertenece al proyecto VINT en el cual participan investigadores de UC Berkeley, LBL, USC/ISI, y Xerox Parc como se describe en [34]. Se encuentra disponible en la pagina <http://www.isi.edu/nsnam/ns/ns-build.html>.

Este simulador fue seleccionado, ya que ha sido ampliamente utilizado para el análisis de nuevos protocolos de red, encaminamiento, colas, tráfico en general, multicast y redes inalámbricas, todo a diferentes niveles: red, transporte y aplicación. Cuenta con un conjunto de "agentes" (clases C++ y Otcl) que nos permiten definir una red con características parecidas a casi cualquier red que encontramos en la actualidad.

NS utiliza dos idiomas porque el simulador tiene dos tipos diferentes de cosas que necesita hacer. Por un lado, las simulaciones detalladas de los protocolos requieren un lenguaje de programación de sistemas que pueden manipular de manera eficiente bytes, cabeceras de paquetes y aplicar los algoritmos que se ejecutan en grandes conjuntos de datos [34].

Por otro lado, una gran parte de la investigación de redes implica ligeramente distintos parámetros o configuraciones, o explorar rápidamente una serie de escenarios. En estos casos, el tiempo de iteración (tiempo para, cambiar el modelo y volver a ejecutar) es más importante [34].

NS cumple con las necesidades con estos dos idiomas, C++ y OTcl. EL primero es rápido para correr pero mas lento para realizar cambio, por lo que es adecuado para la ejecución. OTcl corre mucho más lento, pero se puede cambiar muy rápidamente (y de forma interactiva), lo cual es ideal para la simulación de la configuración [34].



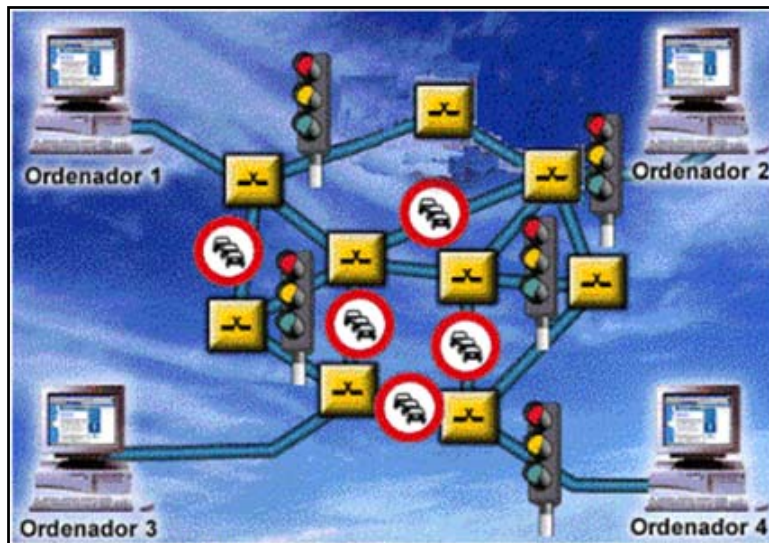
Además, también se han seleccionado las herramientas NAM y Xgraph como complementos, las cuales, aportan mayor información que podría en su momento facilitar a quien le corresponda, una mayor facilidad para la interpretación y análisis del modelo de simulación.

No obstante, una de sus desventajas es la falta de documentación clara y concisa sobre sus diferentes aplicaciones. De lo anterior se deduce que es responsabilidad del autor del presente proyecto, mediante el uso del lenguaje Tcl crear el código del modelo de simulación correspondiente a la red de la UPB seccional Bucaramanga (ver sección 3.4), el cual debe cumplir con las características resaltadas la sección 3.2.

### 3 DESARROLLO DE LA TESIS

#### 3.1 DEFINICIÓN DEL PROBLEMA

Una red de telecomunicaciones es un complejo entramado de recursos físicos y lógicos (medios, equipos de transmisión, conmutadores, protocolos de comunicación, etc.) cuyo objetivo es encausar eficientemente el flujo de información entre los usuarios como se observa en la figura 31.



**Figura 31:** Red de Datos representada como una autopista de información.

Tomado de <http://contenidos.amas.tcorp.telefonica.es>

Es así una cuestión fundamental en el desarrollo de redes de datos el diseñar mecanismos que permitan minimizar el costo de los recursos necesarios para la consecución de una calidad de servicio preestablecida. Tales mecanismos se basan siempre en compartir los

recursos disponibles entre los flujos de los distintos usuarios, de forma que, dada su naturaleza aleatoria, a veces se producen demandas de recursos que no pueden ser atendidas de inmediato.

Con esta perspectiva, los problemas de planificación, operación y mantenimiento de las redes de comunicaciones caen de lleno en el estudio y cabal comprensión de las características de los fenómenos de espera, pues es la base fundamental de la ingeniería de redes.

Un parámetro importante para medir la capacidad de cualquier red (WAN o LAN) es el **ancho de banda**, este se define como la cantidad de información que puede fluir a través de una conexión de red en un período de tiempo [21] .

### 3.1.1 Importancia

Los factores que dan relevancia al buen uso del ancho de banda son los siguientes [21]:

- **El ancho de banda es finito:** El ancho de banda esta limitado, tanto por las leyes de la física como por las tecnologías empleadas para colocar información en el medio.
- **El ancho de banda no es libre:** Es posible comprar equipos para una red LAN que proporcionen un ancho de banda casi ilimitado durante mucho tiempo. Aun así para

las conexiones WAN, casi siempre es necesario comprar ancho de banda a un proveedor de servicios para establecer su conexión a Internet u otras redes.

- **La demanda de ancho de banda es siempre creciente:** En cuanto aparecen nuevas tecnologías e infraestructuras de red, se crean nuevas aplicaciones que se benefician de la mayor capacidad. La distribución por la red del contenido multimedia, incluyendo flujos de datos de vídeo y audio, requieren enormes cantidades de ancho de banda.
- **El ancho de banda es un factor clave para analizar el rendimiento de la red, diseñar nuevas redes y entender Internet.** La información fluye como una cadena de bits desde una computadora a otra a través del mundo, Internet son billones de billones de bits que representan cantidades masivas de información circulando de un lado a otro por todo el globo en segundos, lo cual permite afirmar que en cierto sentido que la capacidad de Internet puede ser medida en términos de banda.

### 3.1.2 Unidad de Ancho de Banda

En los sistemas digitales, la unidad básica del ancho de banda son los bits por segundo (bps). El ancho de banda es la cantidad de información, o bits, que puede fluir de un lugar a otro en un periodo de tiempo específico, o en unos segundos. Aunque el ancho de banda puede describirse en bits por segundo, normalmente se utilizan algunos múltiplos de él

número de bps. Es decir, el ancho de banda de una red normalmente se describe como miles de bits por segundo, millones de bits por segundo o, incluso, billones de bits por segundo. (Ver tabla 5)

<b>UNIDAD DE ANCHO DE BANDA</b>	<b>ABREVIATURA</b>	<b>EQUIVALENTE</b>
Bits por segundo	<b>bps</b>	1 bps = unidad fundamental del ancho de banda
Kilobits por segundo	<b>Kbps</b>	1 kbps = 1000 bps = $10^3$ bps
Megabits por segundo	<b>Mbps</b>	1 Mbps = 1000000 bps = $10^6$ bps
Gigabits por segundo	<b>Gbps</b>	1 Gbps = 1000000000 bps = $10^9$ bps

**Tabla 5** Unidades de ancho de banda.  
Tomada de [21]

### 3.1.3 Limitaciones

El ancho de banda varía en función del tipo de medio de transmisión, así como de las tecnologías LAN y WAN utilizadas. Las diferencias físicas en los caminos de las señales que viajan a través del cobre de par trenzado, fibra óptica e incluso el aire, resultan fundamentales en las limitaciones de la capacidad de transporte de información de un medio dado. Sin embargo, el ancho de banda real de una red está determinado por una combinación del medio físico y las tecnologías elegidas para la señalización de la detección de las señales de la red. El ancho de banda también viene determinado por otros factores variantes, como el número de usuarios en la red, el equipamiento utilizado, las aplicaciones, etc.

Algunos medios de red comunes, junto con sus limitaciones en cuanto a distancia y ancho de banda se muestran en la tabla 6, así mismo los servicios WAN, con su respectivo ancho de banda en la tabla 7.

MEDIO	ANCHO DE BANDA MÁXIMO TEÓRICO	DISTANCIA FÍSICA MÁXIMA
Cable UTP de categoría 5	100 Mbps	100 m
Cable UTP de categoría 5e	Hasta 1000Mbps	100 m
Cable UTP de categoría 6	1Gbps	100 m
Cable UTP de categoría 6a	10 Gpbs	100 m
Fibra óptica multimodo 100Base-FX	100 Mb/ps	2 km
Fibra óptica monomodo o multimodo 1000Base-LX	1000 Mbps	550 km
Fibra óptica multimodo 1000Base-SX	1000 Mbps	550 km
Fibra óptica multimodo 10GBase-SR 10GBase-LX4	1 Gbps	500 km

**Tabla 6:** Limitaciones de longitud y anchos de banda máximos.  
Tomada de [21]

SERVICIOS WAN	USUARIO TÍPICO	ANCHO DE BANDA
Modém	Particulares	56 kbps
ADSL	Particulares, teletrabajadores y pequeñas empresas	512 kbps a 16 Mbps
RDSI	Teletrabajadores y pequeñas empresas	128 kbps
Frame Relay	Pequeñas instituciones y empresas de tamaño medio.	56 kbps a 44,736 Mbps

**Tabla 7:** Servicios WAN comunes y anchos de banda asociados.  
Tomada de [21]

### 3.1.4 Tasa de Transferencia

La cantidad de ancho de banda disponible es una parte crítica de las especificaciones de una red. Puede construirse una LAN típica para proporcionar 100 Mbps a cualquier estación de

trabajo, pero eso no significa que cada usuario pueda mover realmente un centenar de megabits de datos a través de la red en cada segundo de uso. Lo anterior, sólo es cierto bajo las circunstancias más ideales. El concepto de tasa de transferencia de datos puede ayudar a explicar el por qué es así.

La tasa de transferencia (o rendimiento) se refiere al ancho de banda real, medido en un momento concreto del día, empleando rutas concretas de Internet, mientras se transmite un conjunto específico de datos por la red. Desafortunadamente, por muchas razones, la tasa de transferencia es con frecuencia menor que el máximo ancho de banda digital posible del medio que se está empleando. Los siguientes son algunos de los factores que determinan la tasa de transferencia [21]:

- Dispositivos de red.
- Tipos de datos que se van a transferir.
- Topología de la red.
- Número de usuarios en la red
- La computadora del usuario.
- El servidor.
- Congestión.

El ancho de banda teórico de la red es una consideración importante en el diseño de la red, porque la tasa de transferencia de la red nunca es mayor que dicho ancho de banda, debido a las limitaciones impuestas por el medio y las tecnologías de red elegidas.

Los criterios listados anteriormente influyen directamente en las decisiones que un administrador de redes debe tomar, ya que este seleccionará los equipos y servicios a adquirir y en algunos ocasiones estará a su cargo el desarrollo y diseño de la red, para lo cual, deberá anticiparse a la necesidad de incrementar el ancho de banda. En ambos casos, una comprensión del ancho de banda y los cambios en la demanda de este durante un período de tiempo dado, pueden ahorrar a una persona o una empresa una cantidad de dinero significativa.

Aun así y a pesar de su importancia, actualmente los administradores de red calculan el ancho de banda que adquieren del proveedor teniendo como único criterio el ancho de banda que quieren asegurar a cada terminal conectado a la red, lo anterior no tiene en cuenta el comportamiento del usuario, ni un análisis del tráfico previo, lo cual puede llegar a crear deficiencias en el cálculo del mismo, logrando así una red congestionada o por el contrario subutilizada.

***La hipótesis a probar*** mediante el uso del modelo CQN, es que este, puede modelar tráfico HTTP, tendiendo como referencia el control realimentado de congestión de TCP para cada conexión, con el fin de determinar un ancho de banda efectivo (ver sección 2.6.5) para en enlace de la red.

De ahí, que la aplicación de la metodología propuesta como resultado de la evaluación del modelo de red de cola cerrada, será un instrumento a partir del cual los administradores de red podrían tomar decisiones más acertadas.



### 3.2 DESCRIPCIÓN DE LA RED DE DATOS UNIVERSITARIA

Siguiendo la Metodología propuesta en [48] y con el objetivo de obtener información sobre el estado de la red de datos de la UPB seccional Bucaramanga en Mayo de 2006 y en Enero de 2008 de una manera ordenada y eficaz, se tomaron como referencia características de la red que se describen a continuación. Para ello, se describirán los diferentes pasos que se siguieron.

#### 3.2.1 Indagar quien tiene la información

Se debe ubicar el departamento, área u oficina indicada, que tenga a cargo la administración de la red de datos según [48]. Consecuentemente se completaron las siguientes tablas con la información suministrada por la Ing. Sandra Reyes, administradora de la red de datos.

Nombre	Teléfono	E-Mail	Horario De Atención	Cargo	Dpto	Funciones Que Tiene Sobre La Red
Cecilia Amaya	6796220 Ext. Dpto Sistemas	<a href="mailto:camaya@upbbga.edu.co">camaya@upbbga.edu.co</a>	8-12 m y 2-6 p.m.	Jefe Dpto. de Sistemas	Sistemas	Administración Configuración Mantenimiento

**Tabla 8:** Personal encargado de la red de la UPB seccional Bucaramanga en Mayo 2006.  
Tomado de [48]

Nombre	Teléfono	E-Mail	Horario De Atención	Cargo	Dpto	Funciones Que Tiene Sobre La Red
Sandra Pilar Reyes / Edwing Olarte	6796220 Ext. 601	<a href="mailto:spreyes@upbbga.edu.co">spreyes@upbbga.edu.co</a> <a href="mailto:eolarte@upbbga.edu.co">eolarte@upbbga.edu.co</a>	8-12 m y 2-6 p.m.	Administradores Red de datos	Sistemas	Administración Configuración Mantenimiento

**Tabla 9:** Personal encargado de la red de la UPB seccional Bucaramanga en Enero 2008.  
Tomado de [48]

### 3.2.2 Documentar la información según cada elemento de la red

Según [48], se debe indagar sobre cada elemento de la red, y recopilar esta información en diferentes tablas, sin embargo por cuestiones de seguridad la Ing. Sandra Reyes se abstuvo de proporcionar la información de manera detallada.

- **Servidores:** La información descrita en la tabla 10 ha sido tomada de [14], trabajo que se desarrolló paralelamente a la toma de datos para el desarrollo de [2].

NOMBRE	REFERENCIA	PLATAFORMA	SERVICIOS
Abeja	Marca Dell Gama Intermedia	Linux	Correo
Avispa			Servicios Web: (portal principal, páginas de docentes, entre otros)
Garrapata			
Hormiga			
Libélula			
Mosca			
Oruga			
Mantis			
Escarabajo			

**Tabla 10:** Servidores de la red de la UPB seccional Bucaramanga en Mayo de 2006. Tomado de [48] y [14].

- **Switches, Hubs, Enrutadores:** Según [48], se debe realizar un recorrido por las instalaciones para describir la red física y su respectiva ubicación, que facilitará la construcción del plano de la red con el siguiente modelo de tabla:

Dispositivo	Otro-Dispositivo	Modelo Según Marca	En Funcionamiento	Disponible	Sin Conexión
Nombre del dispositivo	Es otro dispositivo conectado al dispositivo principal	Referencia del dispositivo	Puertos conectados al algún elemento o dispositivo	Puertos cableados que No están conectados a algún dispositivo	Puertos que no están cableados.

**Tabla 11 :** Muestra información de dispositivos. Tomado de [48]

Esta información no fue solicitada debido a que su nivel de complejidad no se encuentra dentro de los objetivos del proyecto, sin embargo en [13] describe con detalle la topología de la red UPB seccional Bucaramanga y en [2] describe la topología de la captura de datos que será retomada en esta sección.

- **VLANS:** Una VLAN se puede identificar como una red de área local de carácter virtual; lo que permite al administrador de la red, crear redes lógicas independientes dentro de una sola red física. Las tablas 12 y 13 describen la distribución de las VLAN en la UPB, como se encuentra descrito en [2].

NOMBRE	DIRECCIÓN IP	COMENTARIOS (CANAL)
Vlan 0 Servidores	Ip 172.16.0.X/24	Desde todas las VLANs se puede acceder a Internet a través de los servidores Proxy.
Vlan 2 Administrativa	Ip 172.16.2.X/24	
Vlan 4 Académico	Ip 172.16.4.X/24	
Vlan 6 Nuevas Tecnologías	Ip 172.16.6.X/24	
Vlan 10 Coomeb	Ip 172.16.10.X/24	
Vlan 12 Aulas	Ip 172.16.13.X/24	
Vlan 13 Aulas	Ip 172.16.12.X/24	

**Tabla 12:** VLANS de la red de la UPB seccional Bucaramanga en Mayo de 2006.  
Tomado de [48] y [14]

NOMBRE	DIRECCIÓN IP	COMENTARIOS (CANAL)
1. Default (equipos de red y servidores)	<ul style="list-style-type: none"> <li>• Esta información no se puede suministrar en detalle.</li> <li>• Todo el direccionamiento de la red local es Privado.</li> <li>• Únicamente los servidores tienen acceso a Direcciones IP públicas</li> </ul>	Desde todas las VLANs se puede acceder a Internet a través de los servidores Proxy.
2. Empleados Administrativos		
3. Empleados Académicos		
4. Docentes		
5. Salas de Informática		
6. Laboratorios de clase		
7. Usuarios Móviles		
8. Cooperativa		

**Tabla 13:** VLANS de la red de la UPB seccional Bucaramanga en Enero de 2008.  
Suministrado por Ing. Sandra Reyes

- **Usuarios:** En este aspecto es importante tener claridad de que se nombra como Usuario al total de dispositivos que cuentan con acceso a la red y no a al número de personas que potencialmente pueden usarlos. La distribución de usuarios para los dos enlaces de Internet existentes se encuentra descrita en las tablas 14 y 15.

ENLACE	NÚMERO DE USUARIOS	SEGMENTOS DE RED
Telecom	235	<ul style="list-style-type: none"> <li>• VLANS de estudiantes</li> <li>• ( Salas de Informática, Laboratorios de Clase)</li> </ul>
Impsat	200	<ul style="list-style-type: none"> <li>• VLANS de empleados ( Servidores, Administrativos, Docentes, Académicos y la Cooperativa)</li> </ul>

**Tabla 14:** Distribución de Usuarios en los enlaces de Internet.  
(Mayo de 2006. Tomado de [48])

ENLACE	NÚMERO DE USUARIOS		SEGMENTOS DE RED
	RED FIJA	RED INALÁMBRICA	
Telecom	340	10-100	<ul style="list-style-type: none"> <li>• VLANS de estudiantes</li> <li>• ( Salas de Informática, Laboratorios de Clase)</li> </ul>
Impsat	289	0	<ul style="list-style-type: none"> <li>• VLANS de empleados ( Servidores, Administrativos, Docentes, Académicos y la Cooperativa)</li> </ul>

**Tabla 15:** Distribución de Usuarios en los enlaces de Internet.  
(Enero de 2008. Tomado de [48])

- **Tipo de enlaces:** Se tienen dos canales dedicados Frame Relay cuyas características se encuentran descritas en las tablas 16 y 17.

CANAL	CAPACIDAD	COSTO
Canal TELECOM	2 MB	No hay registro
Canal IMPSAT	2 MB	

**Tabla 16:** Capacidad de cada enlace de la red de la UPB seccional Bucaramanga  
Mayo de 2006, Tomado de [48]

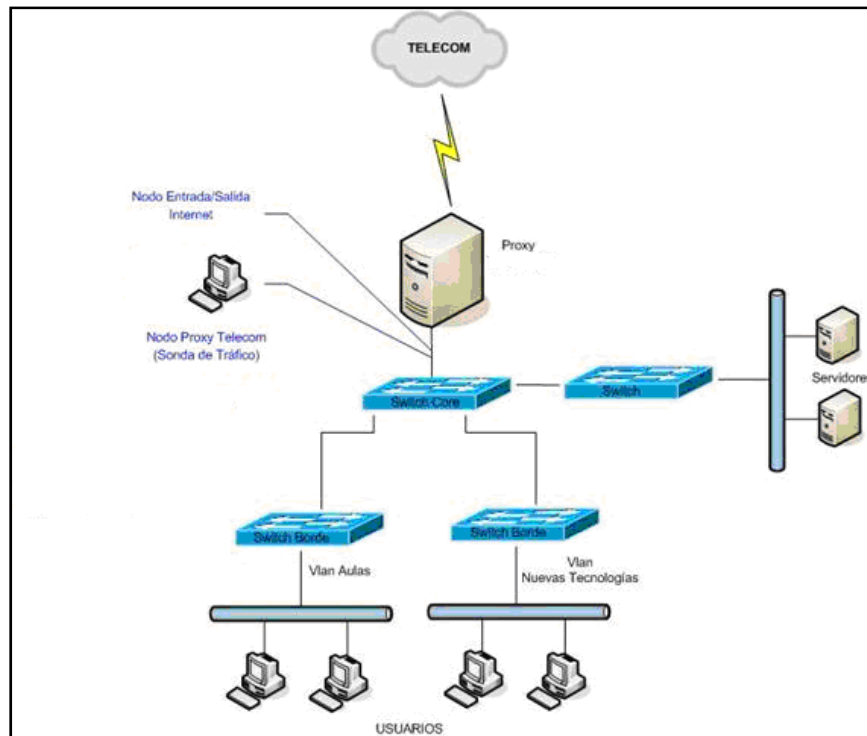
<b>CANAL</b>	<b>CAPACIDAD</b>	<b>COSTO</b>
Canal TELECOM	4 MB	Aproximadamente USD\$ 700 el mega
Canal IMPSAT	4 MB	

**Tabla 17:** Capacidad de cada enlace de la red de la UPB seccional Bucaramanga.  
Enero de 2008, Tomado de [48]

### 3.2.3 Diagrama Lógico de la red de datos UPB seccional Bucaramanga

Como se encuentra descrito en [13], la red tiene una topología de estrella como se observa en la figura 32, que se encuentra dividida en redes de área local virtuales o VLAN que fueron creadas por seguridad y que agrupan las diferentes dependencias de la universidad; las VLAN convergen en el Switch de capa tres que las administra y realiza la tarea de enrutamiento.

El objeto de estudio del presente trabajo es el canal de Telecom, ya que es el que cuenta con la mayor cantidad de usuarios y es el que se proyecta con mayor necesidad de crecimiento ya que mediante este se conectan a Internet las VLAN de Aulas de informática y laboratorios además de los usuarios móviles; los cuales se encuentran al servicio de los estudiantes.



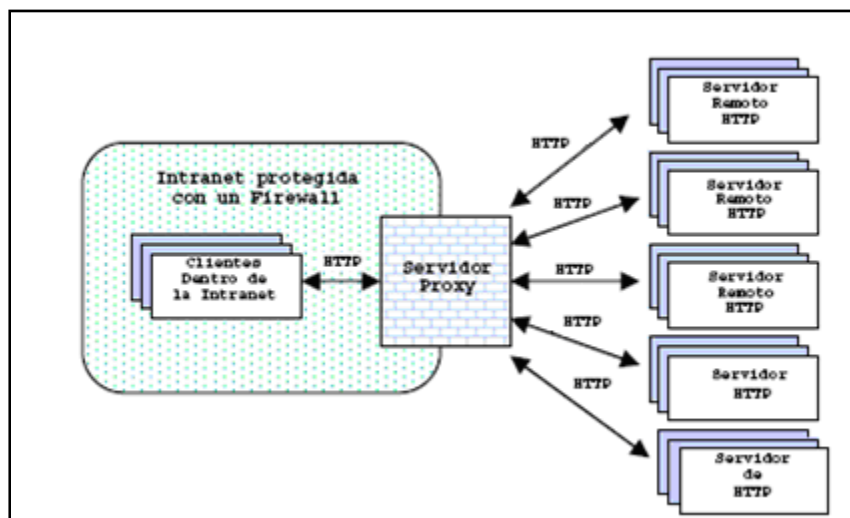
**Figura 32:** Diagrama Lógico de la Red UPB seccional Bucaramanga Canal Telecom. Tomado de [2] y modificado por el autor.

De la figura 32 es posible resaltar que se capturó el tráfico entre la red LAN de la UPB seccional Bucaramanga y el Proxy del canal de Telecom conocido en Mayo de 2006 como Proxy Panda e Identificado con IP 172.16.12.4, por lo cual se obtuvo el flujo de tráfico de la red al Proxy y del Proxy a la red.

Del estudio realizado sobre la red en [2], se puede abstraer que el porcentaje de paquetes UDP comparado con el porcentaje de paquetes TCP es prácticamente despreciable, por tanto en el presente trabajo se ha focalizado la atención sobre el comportamiento del Protocolo de Control de Transmisión (TCP).

De la misma manera, la Ing. Sandra Reyes confirmó que en los datos obtenidos con la sonda ubicada entre el Proxy Panda y la Red de datos UPB seccional Bucaramanga solo se obtuvo tráfico Web o HTTP usando el protocolo TCP, debido a que se tiene un firewall al borde de la red que solo permite este tipo de tráfico.

También es importante resaltar que todo usuario de la red UPB seccional Bucaramanga no se conecta directamente a un servidor Web, debido a que este navega a través del servidor Proxy, ubicado detrás de un cortafuegos (*Firewall*) por cuestiones de seguridad como se observa en la figura 33.



**Figura 33:** Servidor Proxy con Cache.  
Tomado de [42] y modificado por el autor.

El servidor Proxy representa al servidor Web de cara al Usuario y al Usuario de Cara al Servidor, pero este servidor no se limita sólo a emitir peticiones y respuestas del tipo HTTP [42]; el Proxy Panda también se encuentra configurado para actuar como caché, es decir, guardar de manera local los documentos que han sido mas recientemente o frecuentemente

consultados, para ser usados en futuras peticiones, evitando así tener que solicitarlos a los servidores Web respectivos, mejorando la velocidad de acceso y disminuyendo el tráfico de la red, en este caso disminuyendo el tráfico a través del canal Telecom. Es importante aclarar que debido a la ubicación de la sonda de tráfico, la captura muestra las tramas de datos de la red hacia el Proxy y del Proxy hacia la red, por tanto el efecto del caché se verá reflejado en el promedio del tiempo de transferencia de los objetos HTTP.

De acuerdo a lo que se ha descrito del tráfico de la red obtenido y analizado en [2] se determino que los protocolos objeto de investigación en el presente trabajo son TCP y HTTP los cuales se encuentran descritos en las secciones 2.4 y 2.5 respectivamente.

### **3.2.4 Restricciones administrativas sobre la red**

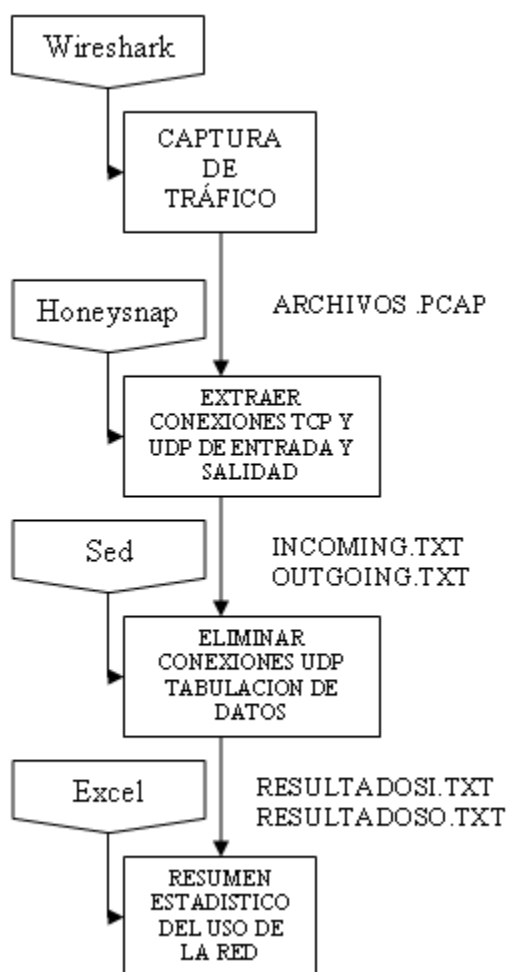
Más allá de los aspectos físicos, la red requiere un cierto grado de seguridad. El primer nivel de protección es generalmente un cortafuego (*firewall*) en el punto de acceso del Internet y a continuación encontramos el Servidor *Proxy*. Las reglas y los filtros del Proxy se deben definir basados en las necesidades del acceso de red de la organización. Las políticas administrativas sobre la red se listan a continuación:

- El cortafuegos (*firewall*) se utiliza para proteger las redes internas en una manera restrictiva, teniendo un direccionamiento local de rango privado.
- La filtración y las reglas en el *Proxy* permiten solamente conexiones salientes de tráfico HTTP, a los servidores del correo y a los servidores de nombres (DNS) permitidos.



### 3.3 CARACTERIZACIÓN DE LAS TRAZAS DE TRÁFICO

Con el conocimiento de los parámetros necesarios para evaluar el modelo (ver sección 2.6.4), se analizaron y caracterizaron las trazas de tráfico tomadas de la red de datos de la Universidad Pontificia Bolivariana seccional Bucaramanga en [2], mediante los pasos de la figura 34, que serán descritos en esta sección.



**Figura 34:** Pasos para la caracterización de las trazas de tráfico.  
Creado por el autor.

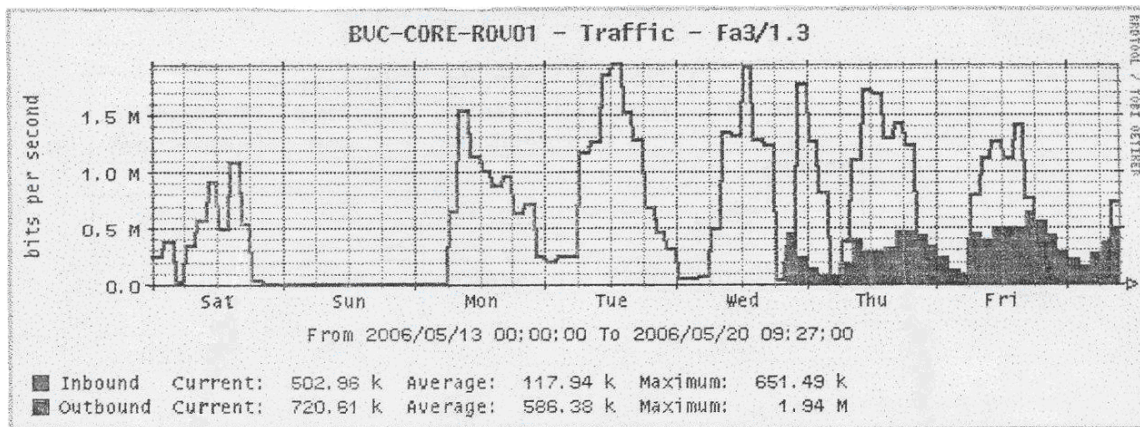
### 3.3.1 Medición del desempeño de la red

Cuando una red tiene un mal desempeño, los usuarios frecuentemente se quejan con los administradores de la red, exigiendo mejoras. Para mejorar el desempeño, los operadores deben primero determinar con exactitud la causa del problema. Con el fin de determinarla, se deben efectuar mediciones (para realizar una captura con Wireshark ver sección 2.7.1). Para lo cual se deben seguir las siguientes recomendaciones [38]:

- **El tamaño de la muestra debe ser grande:** Una muestra grande reducirá la incertidumbre de la media y de la desviación estándar medidas.
- **La muestra debe ser representativa:** Las mediciones deberían repetirse a diferentes horas del día y de la semana para observar el efecto de diferentes cargas sobre el sistema.
- **La muestra debe ser en condiciones normales:** Tomar la muestra sobre la red, el día en que se presenta un acontecimiento especial, podría dar resultados que no son representativos.
- **Tener en cuenta el reloj del sistema:** Los relojes de computador funcionan sumando uno a uno a un contador a intervalos regulares. El intervalo del analizador de protocolos Wireshark es de 1 segundo. En consecuencia, toda transacción con duración menor a un segundo, tendrá un error implícito en la medición.
- **Tener en cuenta el efecto del caché:** A menos que se quiera medir el desempeño de la red teniendo en cuenta los efectos del caché sobre el tiempo de transferencias de archivos, se deben tomar medidas sobre el sistema inactivo creando una carga de tráfico tal que se produzca desbordamiento del caché.

### 3.3.2 Selección de la muestra

En las observaciones de [13], se resalta como una ventaja para el administrador de la red el uso de la herramienta MRTG con el fin de determinar el tráfico entrante y saliente así como el ancho de banda de una interfase. Por tanto para la selección de la muestra se tuvo en cuenta el gráfico del MRTG (véase figura 35) de la tercera semana del Mes de Mayo de 2006. Del trabajo [2], donde se observa el pico de mayor tráfico para el enlace Telecom hacia la red el martes y miércoles, 16 y 17 de Mayo respectivamente, lo cual nos daría una red congestionada para esos períodos.



**Figura 35:** MRTG Canal Telecom Tercera Semana de Mayo de 2006.  
Tomada de [2].

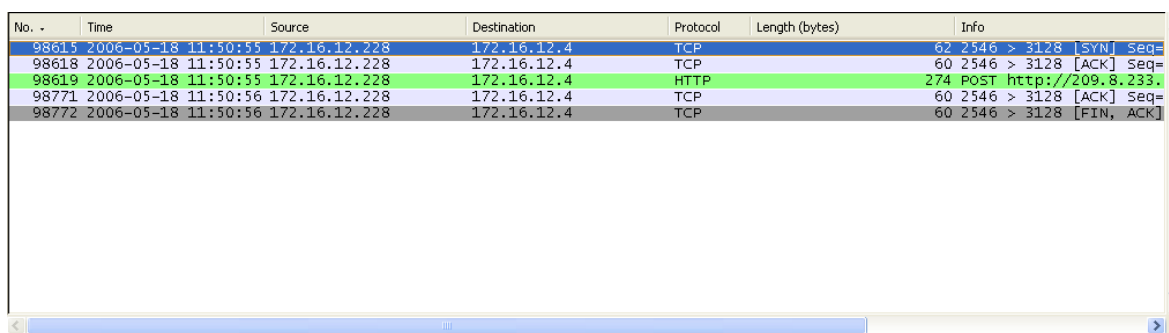
Sin embargo, en el mismo trabajo se establece que la toma de datos continua sobre el canal de Telecom no se efectuó en esas fechas, por lo anterior, se procedió a seleccionar el Jueves, Viernes y Sábado, 18, 19 y 20 de Mayo de 2006 respectivamente como se observan en la tabla 35. En este caso se tiene una muestra de tres días, con 348 archivos, cada uno de 50MB, lo cual nos da un total de 17.4GB de información.

CD	Sitio de prueba	Rango de tiempo (día, Mes, año)	Archivos .pcap	Tamaño de cada archivo .pcap	Archivos de tráfico ato aprox.	Tiempo de máxima intensidad de tráfico.	Archivo .pcap con máximo tráfico
8	Telecom	180506 09:39:26 A.m. 10:15:48 A.m.	1-9	50 Mb	1-6	9:39:36 A.m. 10:05 A.m.	4
9	Telecom	180506 10:18:24 A.m. 2:14:56 P.m.	10-89	50 Mb	15-89	10:34 A.m. 2:14 P.m.	15
10	Telecom	180506 2:19:08 P.m. 5:33:48 P.m.	90-147	50 Mb	90-111	2:19 P.m. 3:16 P.m.	90
11	Telecom	190506 8:33:16 A.m. 2:28:14 P.m.	1-82	50 Mb	26-82	10:21:36 A.m. 2:28:14 P.m.	45
12	Telecom	190506 2:31:10 A.m. 8:25:54 P.m. 200506 6:34:56 A.m. 11:16:06 A.m.	83-135 136-170	50 Mb 50Mb	95-135 136-160	3:16:06 A.m. 8:25:54 P.m. 6:34:06 A.m. 10:29:00 A.m.	95 136
13	Telecom	200506 11:19:10 A.m. 06:28:44 P.m.	171-205	50 Mb	184-205	1:24:10 P.m. 6:28:44 P.m.	203

**Tabla 18:** Distribución de los archivos de la muestra seleccionada.  
Tomado de [2].

### 3.3.3 Obtención de trazas TCP con Wireshark

Una conexión TCP equivale a un objeto si el protocolo es HTTP 1.0 (véase sección 2.5.2). Tomando una conexión TCP del 18 de Mayo, archivo 41, con filtro “p.dst==172.16.12.4 && tcp.dstport==3128 && ip.src==172.16.12.228 && tcp.srcport==2546” en Wireshark se tiene el resultado de la figura 36:



The screenshot shows a packet capture in Wireshark with the following data:

No. -	Time	Source	Destination	Protocol	Length (bytes)	Info
98615	2006-05-18 11:50:55	172.16.12.228	172.16.12.4	TCP	62	2546 > 3128 [SYN] Seq=
98618	2006-05-18 11:50:55	172.16.12.228	172.16.12.4	TCP	60	2546 > 3128 [ACK] Seq=
98619	2006-05-18 11:50:55	172.16.12.228	172.16.12.4	HTTP	274	POST http://209.8.233.
98771	2006-05-18 11:50:56	172.16.12.228	172.16.12.4	TCP	60	2546 > 3128 [ACK] Seq=
98772	2006-05-18 11:50:56	172.16.12.228	172.16.12.4	TCP	60	2546 > 3128 [FIN, ACK]

**Figura 36:** Conexión TCP en Wireshark.  
Creado por el autor.

Como se observa en esta figura, hay 5 paquetes con tamaño de 516 bytes, esto incluye el tamaño de archivo, y los encabezados, más el valor de los 4 paquetes correspondientes al saludo de tres vías de TCP.

### 3.3.4 Extracción de Conexiones TCP y UDP con Honeysnap.

Tomando como base el archivo de ejemplo \$Id: honeynet.cfg 4936 2006-12-15 10:39:22Z desarrollado por Arthur Clune. Este se editó de tal manera que se pudiera obtener un resumen del archivo pcap analizado solo, las conexiones TCP desde y hacia el Proxy Panda

identificado con la IP 172.16.12.4. Para continuar, se digitó: *honeysnap-c honeynet.cfg file.pcap*, en la línea de comandos, como se registra en la figura 37.

```

C:\Documents and Settings\Xinena>cd..
C:\Documents and Settings>cd..
C:\>cd Python24
C:\Python24>cd Scripts
C:\Python24\Scripts>honeysnap -c honeynet.cfg 20_136 20_137 20_138 20_139 20_140
20_141 20_142 20_143 20_144 20_145 20_146 20_147 20_148 20_149 20_150 20_151 20
152 20_153 20_154 20_155 20_156 20_157 20_158 20_159 20_160 20_161 20_162 20_16
3 20_164 20_165 20_166 20_167 20_168 20_169 20_170 20_171 20_172 20_173 20_174 2
0_175 20_176 20_177 20_178 20_179 20_180 20_181 20_182 20_183 20_184 20_185 20_1
86 20_187 20_188 20_189 20_190 20_191 20_192 20_193 20_194 20_195 20_196 20_197
20_198 20_199 20_200 20_201 20_202 20_203 20_204 20_205

```

**Figura 37:** Ejecutando Honeysnap en la línea de comandos.  
Creada por el autor.

Finalmente se obtienen dos archivos de texto plano:

- El archivo de *incoming.txt* contiene las conexiones TCP y UDP de la red UPB seccional Bucaramanga hacia el Proxy Panda, como se observa en la figura 38.

Incoming connections for 172.16.12.4

TCP TRAFFIC SUMMARY:

Start	End	Source	Sport	Dest	Dport	Count	Bytes
Fri May 19 20:27:18 2006	Fri May 19 20:29:29 2006	172.16.12.241	1834	172.16.12.4	3128	83	17433
Sat May 20 03:07:43 2006	Sat May 20 03:07:43 2006	172.16.6.5	1086	172.16.12.4	53427	1	0
Sat May 20 00:52:43 2006	Sat May 20 00:52:43 2006	172.16.6.5	1086	172.16.12.4	51648	1	0
Sat May 20 00:02:43 2006	Sat May 20 00:02:43 2006	172.16.6.5	1086	172.16.12.4	50989	1	0
Sat May 20 03:13:43 2006	Sat May 20 03:13:43 2006	172.16.6.5	1086	172.16.12.4	53506	1	0
Fri May 19 21:08:43 2006	Fri May 19 21:08:43 2006	172.16.6.5	1086	172.16.12.4	48630	1	0
Fri May 19 22:05:43 2006	Fri May 19 22:05:43 2006	172.16.6.5	1086	172.16.12.4	49384	1	0
Sat May 20 01:31:43 2006	Sat May 20 01:31:43 2006	172.16.6.5	1086	172.16.12.4	52162	1	0
Sat May 20 00:41:43 2006	Sat May 20 00:41:43 2006	172.16.6.5	1086	172.16.12.4	51503	1	0
Sat May 20 04:27:43 2006	Sat May 20 04:27:43 2006	172.16.6.5	1086	172.16.12.4	54490	1	0
Fri May 19 20:47:43 2006	Fri May 19 20:47:43 2006	172.16.6.5	1086	172.16.12.4	48286	1	0
Sat May 20 05:18:43 2006	Sat May 20 05:18:43 2006	172.16.6.5	1086	172.16.12.4	55163	1	0
Fri May 19 20:26:25 2006	Fri May 19 20:27:17 2006	172.16.12.241	1830	172.16.12.4	3128	10	1610

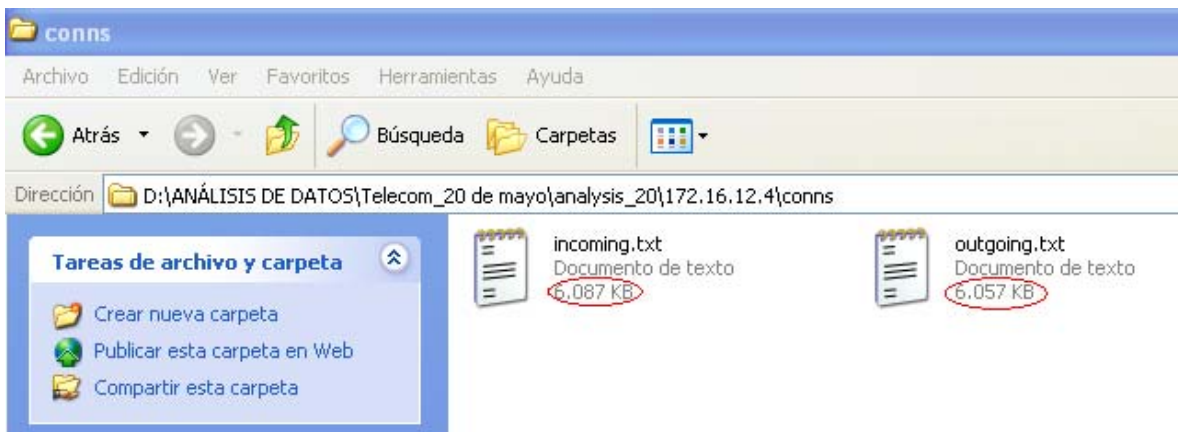
**Figura 38:** Resumen de Conexiones TCP archivo incoming.txt.  
Creado por el autor.

- El archivo de *outgoing.txt* contiene las conexiones TCP y UDP del Proxy Panda hacia la red UPB seccional Bucaramanga como se observa en la figura 397.

Start	End	Source	Sport	Dest	Dport	Count	Bytes
Sat May 20 06:34:36 2006	Sat May 20 06:34:48 2006	172.16.12.4	47887	172.16.0.131	53	5	179
Sat May 20 04:04:33 2006	Sat May 20 04:04:33 2006	172.16.12.4	47846	172.16.0.131	53	1	45
Sat May 20 04:04:34 2006	Sat May 20 06:32:57 2006	172.16.12.4	47882	172.16.0.131	53	92	3609
Sat May 20 04:04:33 2006	Sat May 20 04:04:33 2006	172.16.12.4	47845	172.16.0.131	53	4	175
Sat May 20 04:04:34 2006	Sat May 20 04:04:34 2006	172.16.12.4	47881	172.16.0.131	53	1	42
Sat May 20 04:04:33 2006	Sat May 20 04:04:33 2006	172.16.12.4	47840	172.16.0.131	53	1	43
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47739	172.16.0.131	53	1	45
Sat May 20 04:04:27 2006	Sat May 20 04:04:27 2006	172.16.12.4	47775	172.16.0.131	53	5	220
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47734	172.16.0.131	53	1	46
Sat May 20 04:04:26 2006	Sat May 20 04:04:26 2006	172.16.12.4	47770	172.16.0.131	53	6	263
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47733	172.16.0.131	53	1	43
Sat May 20 04:04:26 2006	Sat May 20 04:04:26 2006	172.16.12.4	47769	172.16.0.131	53	1	45
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47728	172.16.0.131	53	1	43
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47764	172.16.0.131	53	1	44
Sat May 20 04:04:25 2006	Sat May 20 04:04:25 2006	172.16.12.4	47763	172.16.0.131	53	1	43
Sat May 20 02:40:40 2006	Sat May 20 03:39:28 2006	172.16.12.4	47722	172.16.0.131	53	1291	59212

**Figura 39:** Resumen de Conexiones UDP archivo outgoing.txt.  
Creado Por el autor.

Es importante resaltar que si tomamos el ejemplo de la figura 37 y lo confrontamos con la tabla 18, se observa que se analizaron los archivos correspondientes al 20 de Mayo, para un total de 68 archivos, con un espacio en disco de 2.9 GB, los cuales fueron reducidos (véase sección 2.7.5) a dos archivos con un espacio en disto de 12MB como se observa en la figura 40.



**Figura 40:** Archivos generados por Honeysnap.  
Creado Por el autor.

Retomando la conexión TCP analizada con Wireshark (véase figura 36) del archivo de *outgoing.txt* correspondiente al 18 de Mayo. Aun se tienen los cinco paquetes, pero el tamaño total es de 220 bytes (una diferencia de 296 bytes), la razón es muy simple, con Honeysnap se obtiene la cantidad de bytes de datos, es decir, del tamaño del objeto transferido.

The screenshot shows a Notepad window with the following content:

```

Thu May 18 11:47:47 2006 Thu May 18 11:47:47 2006 172.16.0.131 53 172.16.12.4 36123 1 261
Incoming connections for 172.16.12.4
TCP TRAFFIC SUMMARY:
Start End Source Sport Dest Dport Count Bytes
Thu May 18 11:50:55 2006 Thu May 18 11:50:56 2006 172.16.12.228 2546 172.16.12.4 3128 5 220
Thu May 18 11:48:53 2006 Thu May 18 11:48:54 2006 172.16.6.5 55805 172.16.12.4 54629 3 0
Thu May 18 11:49:46 2006 Thu May 18 11:50:49 2006 172.16.12.12 2282 172.16.12.4 3128 6 1348
Thu May 18 11:50:15 2006 Thu May 18 11:50:15 2006 172.16.6.5 55805 172.16.12.4 55984 2 0
Thu May 18 11:49:36 2006 Thu May 18 11:49:36 2006 172.16.6.5 55805 172.16.12.4 55257 2 0
Thu May 18 11:50:27 2006 Thu May 18 11:50:27 2006 172.16.6.5 55805 172.16.12.4 56182 2 0
Thu May 18 11:48:57 2006 Thu May 18 11:48:57 2006 172.16.0.142 80 172.16.12.4 54670 4 242
Thu May 18 11:50:51 2006 Thu May 18 11:50:53 2006 172.16.13.22 1187 172.16.12.4 3128 7 1429

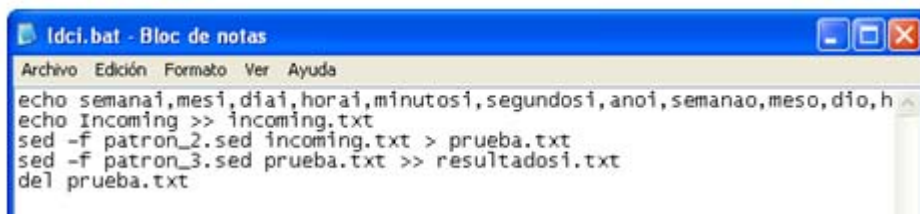
```

**Figura 41:** Conexión TCP con Honeysnap.  
Creado por el autor.

### 3.3.5 Organización de la información con Sed

Los archivos “.bat” o archivos por lotes, tienen comandos que son ejecutados en grupo sobre la línea de comando, permitiendo automatizar la tarea, tal como se observa en la figura 30. Donde, el comando “echo”, imprime un texto en la pantalla, además si se encuentra escrito como “echo Texto > Nombre.extension” se creará un archivo con el nombre y la extensión relacionadas después del símbolo “>”. Por otra parte si se encuentra escrito como “echo Texto >> Nombre.extension” se escribirá en la siguiente línea en blanco que tenga el archivo relacionado. De la misma manera se incluyen las líneas de comando para ejecutar Sed.

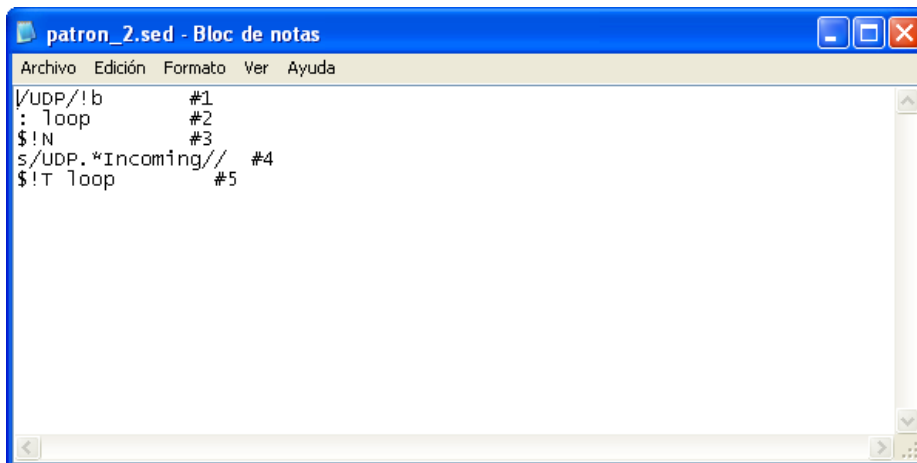




```
ldci.bat - Bloc de notas
Archivo Edición Formato Ver Ayuda
echo semana1,mes1,dia1,hora1,minutos1,segundos1,ano1,semanao,meso,dio,h
echo Incoming >> incoming.txt
sed -f patron_2.sed incoming.txt > prueba.txt
sed -f patron_3.sed prueba.txt >> resultados1.txt
del prueba.txt
```

**Figura 42:** Línea de Comandos ldci.bat para editar incoming.txt.  
Creado por el autor.

Mediante las instrucciones del archivo “.bat” de la figura 30, se crea un archivo resultados1.txt, cuya primera línea, es el encabezado de los datos. Se escribe la palabra *Incoming* en la última línea en Blanco del archivo *incoming.txt*. Se usa el comando “-f patron\_2.sed” (véase sección 2.7.3) sobre el archivo *incoming.txt*, el cual borra todas las líneas que se encuentren entre la Palabra *UDP* y la Palabra *Incoming*. A continuación se usa el comando “-f patron\_3.sed” (véase figura 43), el cual elimina todas las líneas en blanco, todas las líneas que empiecen con las palabras “start”, “tcp” e “Incoming”. Además, también elimina cualquier espacio entre caracteres y lo cambiará por una “,”.



```
patron_2.sed - Bloc de notas
Archivo Edición Formato Ver Ayuda
[/UDP/!b      #1
: loop       #2
$!N         #3
s/UDP.*Incoming// #4
$!T loop    #5
```

**Figura 43:** Archivo Patron\_2.sed.  
Creado por el autor.

Finalmente se obtienen dos archivos, el primero llamado *resultadosi.txt* (ver figura 44 ) y el segundo llamado *resultadoso.txt* (ver figura 45), con r tulos y columnas separadas por comas para que puedan ser f cilmente analizadas en Excel.

```

resultadosi.txt - Bloc de notas
Archivo Edici3n Formato Ver Ayuda
inicio1, inicio2, fin1, fin2, ips1, ips2, ips3, ips4, puertos, ipd1, ipd2, ipd3, ipd4, puertod, cantidad, bytes
1147971055, 8143, 1147971056, 0804, 172, 16, 12, 228, 2546, 172, 16, 12, 4, 3128, 5, 220
1147970933, 7490, 1147970934, 0106, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 54629, 3, 0
1147970986, 4146, 1147971049, 6599, 172, 16, 12, 12, 2282, 172, 16, 12, 4, 3128, 6, 1348
1147971015, 2120, 1147971015, 2304, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 55984, 2, 0
1147970976, 9505, 1147970976, 9657, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 55257, 2, 0
1147971027, 1463, 1147971027, 1627, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 56182, 2, 0
1147970937, 1620, 1147970937, 1656, 172, 16, 0, 142, 80, 172, 16, 12, 4, 54670, 4, 242
1147971051, 1254, 1147971053, 6665, 172, 16, 13, 22, 1187, 172, 16, 12, 4, 3128, 7, 1429
1147971000, 7359, 1147971000, 7512, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 55688, 2, 0
1147970921, 9712, 1147970921, 9740, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 54468, 2, 0
1147970901, 2676, 1147970901, 5305, 172, 16, 12, 231, 1844, 172, 16, 12, 4, 3128, 5, 192
1147970950, 9335, 1147970950, 9359, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 54899, 2, 0
1147971002, 7998, 1147971002, 8005, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 55737, 2, 0
1147970978, 5378, 1147970978, 8276, 172, 16, 12, 479, 2256, 172, 16, 12, 4, 3128, 6, 1250
1147971015, 9778, 1147971015, 9963, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 56015, 2, 0
1147970982, 5034, 1147970982, 5281, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 55330, 2, 0
1147970900, 8621, 1147970901, 1894, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 54142, 3, 0
1147970993, 6286, 1147970993, 6484, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 55537, 2, 0
1147970954, 4905, 1147970954, 4911, 172, 16, 6, 5, 55805, 172, 16, 12, 4, 54948, 2, 0
1147970918, 8879, 1147970921, 3444, 172, 16, 13, 22, 1163, 172, 16, 12, 4, 3128, 12, 3195

```

**Figura 44:** Resultadosi.txt obtenidos con Sed Creado por el autor.

```

resultadoso.txt - Bloc de notas
Archivo Edici3n Formato Ver Ayuda
inicio1, inicio2, fin1, fin2, ips1, ips2, ips3, ips4, puertos, ipd1, ipd2, ipd3, ipd4, puertod, cantidad, bytes
1147970980, 7979, 1147970984, 0071, 172, 16, 12, 4, 3128, 172, 16, 12, 231, 1885, 5, 140
1147971028, 5121, 1147971028, 5123, 172, 16, 12, 4, 56197, 172, 16, 6, 5, 55805, 3, 12
1147970975, 7837, 1147970975, 7844, 172, 16, 12, 4, 55224, 172, 16, 6, 5, 55805, 3, 12
1147971012, 4604, 1147971012, 4605, 172, 16, 12, 4, 55914, 172, 16, 6, 5, 55805, 3, 12
1147970906, 7630, 1147970906, 7656, 172, 16, 12, 4, 54207, 172, 16, 6, 5, 55805, 3, 12
1147970998, 6848, 1147970998, 6869, 172, 16, 12, 4, 55646, 172, 16, 6, 5, 55805, 3, 12
1147971034, 7999, 1147971034, 8001, 172, 16, 12, 4, 56328, 172, 16, 6, 5, 55805, 3, 12
1147970933, 5625, 1147970933, 5629, 172, 16, 12, 4, 54621, 172, 16, 6, 5, 55805, 3, 12
1147970975, 1306, 1147970978, 0545, 172, 16, 12, 4, 3128, 172, 16, 12, 12, 2278, 9, 4420
1147970909, 7919, 1147970909, 7920, 172, 16, 12, 4, 54274, 172, 16, 6, 5, 55805, 3, 12
1147970944, 3894, 1147970946, 0459, 172, 16, 12, 4, 3128, 172, 16, 12, 44, 2582, 5, 367
1147970993, 8894, 1147970993, 8914, 172, 16, 12, 4, 55548, 172, 16, 6, 5, 55805, 3, 12
1147971031, 7527, 1147971032, 0834, 172, 16, 12, 4, 3128, 172, 16, 12, 228, 2516, 5, 140
1147971031, 3998, 1147971031, 4000, 172, 16, 12, 4, 56237, 172, 16, 6, 5, 55805, 3, 12
1147970953, 7727, 1147970953, 7753, 172, 16, 12, 4, 3128, 172, 16, 12, 30, 3118, 2, 0
1147970911, 8333, 1147970912, 1439, 172, 16, 12, 4, 3128, 172, 16, 12, 203, 2236, 6, 91
1147971033, 4755, 1147971046, 9745, 172, 16, 12, 4, 3128, 172, 16, 12, 164, 2398, 13, 2133

```

**Figura 45:** Resultadoso.txt obtenidos con Sed Creado por el autor.

### 3.3.6 Resultados de la utilizaci3n del canal de Internet con Excel.

En el documento [2] se hab a realizado un an lisis estad stico sobre el uso de la red UPB seccional Bucaramanga. No obstante este se realizo al nivel de red y sobre archivos de 3 minutos, por esta raz3n se proyectaron los siguientes an lisis, correspondientes al nivel de

aplicación (tamaño de objetos) y teniendo en cuenta el tráfico de todo el día (no solo de los tres minutos de la hora pico).

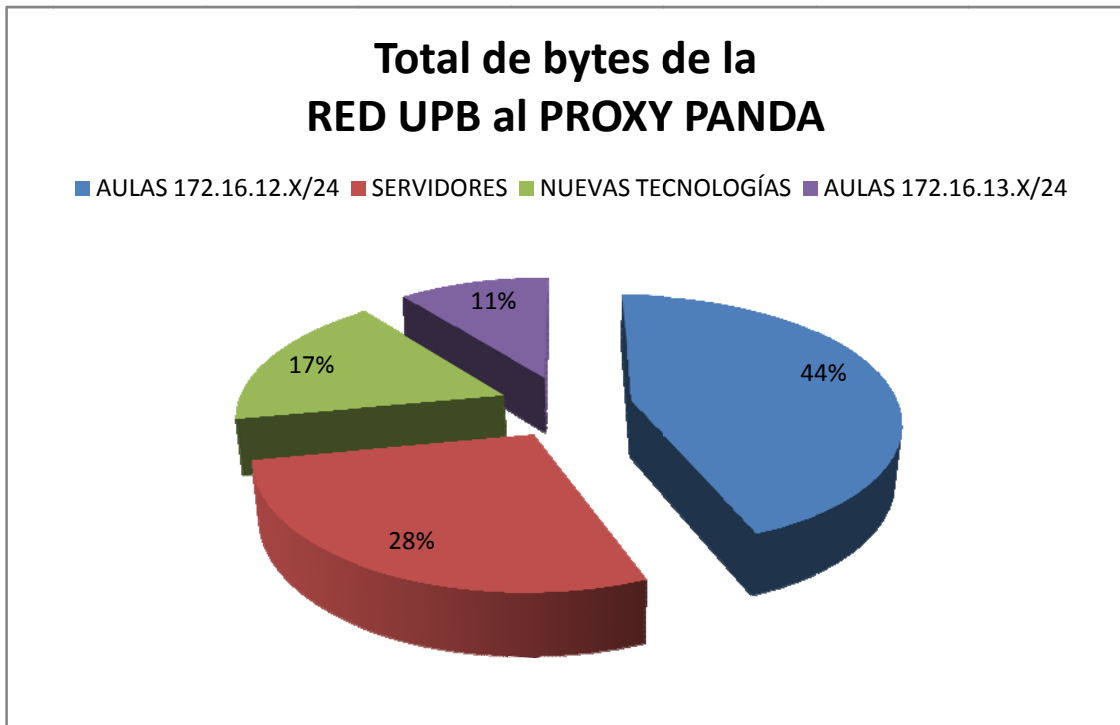
Estos análisis se realizaron importando los archivos de texto generados por Sed, al software de aplicación Excel 2007 (debido a que el número de filas (conexiones TCP) excede la capacidad que puede manejar Excel 2003). Posteriormente se aplicó sobre la información tabulada una tabla dinámica que permite seleccionar de manera eficaz la información expuesta a continuación.

- **Análisis Estadístico del 18 de Mayo de 2006**

El 18 de Mayo de 2006, se encontraron 201 usuarios intercambiando 164006 objetos mediante conexiones TCP de la red UPB seccional Bucaramanga al Proxy Panda. Como se observa en la Tabla 19 y en la Figura 46.

<b>VLANs</b>	<b>Porcentaje de bytes</b>	<b>Total bytes</b>
AULAS 172.16.12.X/24	44.01%	241669500
SERVIDORES	27.98%	153663511
NUEVAS TECNOLOGÍAS	17.34%	95203376
AULAS 172.16.13.X/24	10.67%	58573058
<b>Total general</b>	<b>100.00%</b>	<b>549109445</b>

**Tabla 19:** Total de bytes de la Red UPB al Proxy Panda el 18 de Mayo de 2006.  
Creada por el autor.

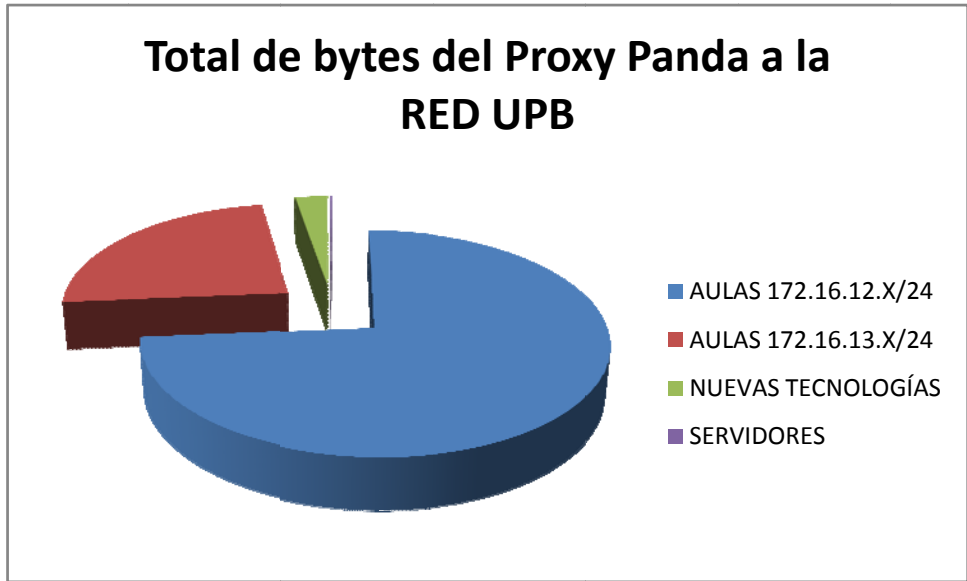


**Figura 46:** Total de bytes de la Red UPB al Proxy Panda el 18 de Mayo de 2006.  
Creada por el autor.

De la misma manera, el 18 de Mayo de 2006 se encontraron 199 usuarios intercambiando 163378 objetos mediante conexiones TCP del Proxy Panda a la red UPB seccional Bucaramanga. Como se Observa en la Tabla 20 y en la Figura 47.

VLANS	Porcentaje de bytes	Total bytes
AULAS 172.16.12.X/24	73.63%	2128680109
AULAS 172.16.13.X/24	23.73%	686139072
NUEVAS TECNOLOGÍAS	2.61%	75462350
SERVIDORES	0.03%	927859
<b>Total general</b>	<b>100.00%</b>	<b>2891209390</b>

**Tabla 20:** Total de bytes del Proxy Panda a la Red UPB el 18 de Mayo de 2006.  
Creada por el autor.



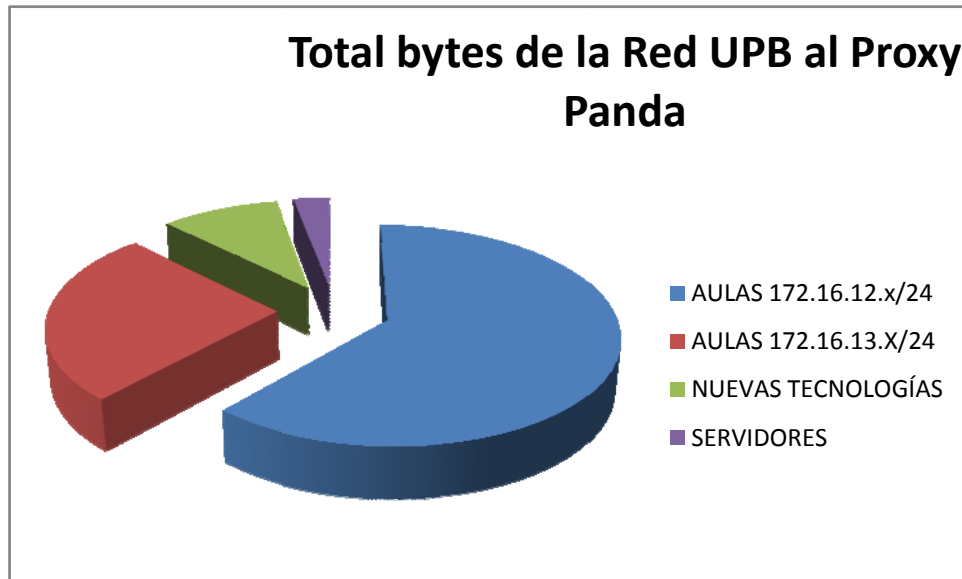
**Figura 47:** Total de bytes de Proxy Panda a la Red UPB el 18 de Mayo de 2006.  
Creada por el autor.

- Análisis Estadístico del 19 de Mayo de 2006**

El 19 de Mayo de 2006, se encontraron 233 usuarios intercambiando 208985 archivos u objetos mediante conexiones TCP de la red UPB seccional Bucaramanga al Proxy Panda, como se Observa en la Tabla 21 y en la Figura 48.

VLANS	Porcentaje de bytes	Total bytes
AULAS 172.16.12.X/24	61.57%	491740756
AULAS 172.16.13.X/24	25.89%	206800670
NUEVAS TECNOLOGÍAS	9.63%	23152185
SERVIDORES	2.90%	76918778
<b>Total general</b>	<b>100.00%</b>	<b>798612389</b>

**Tabla 21:** Total de bytes de la Red UPB al Proxy Panda el 19 de Mayo de 2006.  
Creada por el autor.

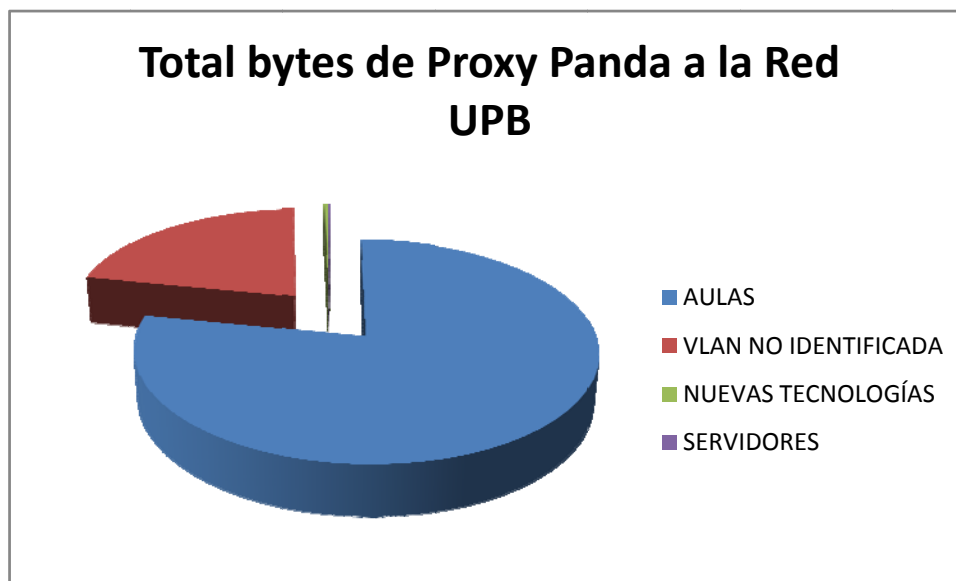


**Figura 48:** Total de bytes de la Red UPB al Proxy Panda el 19 de Mayo de 2006.  
Creada por el autor.

De la misma manera, el 19 de Mayo de 2006 se encontraron 234 usuarios intercambiando 207507 archivos u objetos mediante conexiones TCP del Proxy Panda a la red UPB seccional Bucaramanga, tal como se Observa en la Tabla 22 y en la Figura 49.

VLANS	Porcentaje de bytes	Total bytes
AULAS 172.16.12.X/24	78.17%	4261210646
AULAS 172.16.13.X/24	21.44%	1168652477
NUEVAS TECNOLOGÍAS	0.34%	18353702
SERVIDORES	0.05%	2744649
<b>Total general</b>	<b>100.00%</b>	<b>5450961474</b>

**Tabla 22:** Total de bytes del Proxy Panda ala Red UPB el 19 de Mayo de 2006.  
Creada por el autor.



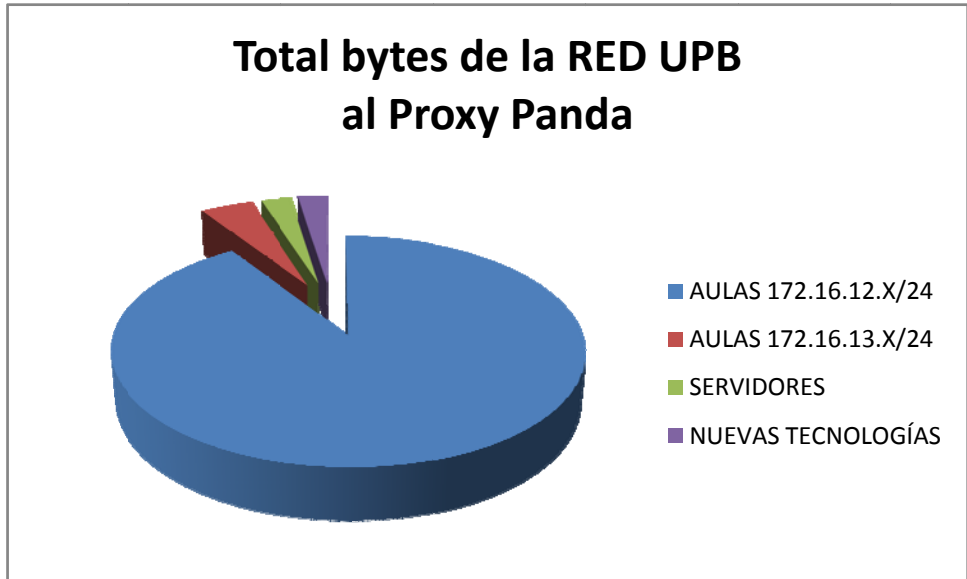
**Figura 49:** Total de bytes de Proxy Panda a la Red UPB el 19 de Mayo de 2006.  
Creada por el autor.

- **Análisis Estadístico del 20 de Mayo de 2006**

El 20 de Mayo de 2006 se encontraron 121 usuarios intercambiando 45847 archivos u objetos mediante conexiones TCP de la red UPB seccional Bucaramanga al Proxy Panda, como se Observa en la Tabla 23 y en la Figura 50.

VLANs	Porcentaje de bytes	Total bytes
AULAS 172.16.12.X/24	90.81%	521507203
AULAS 172.16.13.X/24	4.30%	24716141
NUEVAS TECNOLOGÍAS	2.47%	13915894
SERVIDORES	2.42%	14161080
<b>Total general</b>	<b>100.00%</b>	<b>574300318</b>

**Tabla 23:** Total de bytes de la Red UPB al Proxy Panda el 20 de Mayo de 2006.  
Creada por el autor.



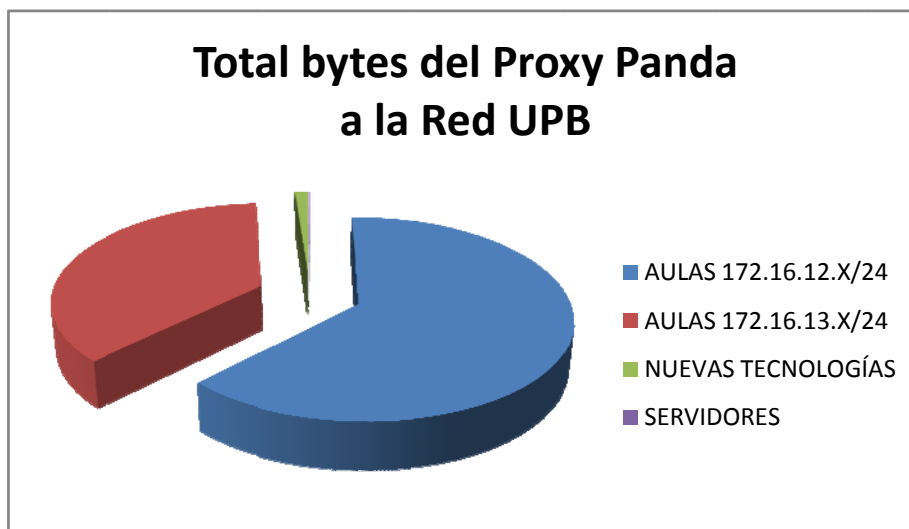
**Figura 50:** Total de bytes de la Red UPB al Proxy Panda el 20 de Mayo de 2006.  
Creada por el autor.

De la misma manera el 20 de Mayo de 2006, se encontraron 121 usuarios intercambiando 45626 archivos u objetos mediante conexiones TCP del Proxy Panda a la red UPB seccional Bucaramanga, como se Observa en la Tabla 24 y en la Figura 51.

VLANs	Porcentaje de bytes	Total bytes
AULAS 172.16.12.X/24	62.40%	1649593624
AULAS 172.16.13.X/24	36.48%	964477849
NUEVAS TECNOLOGÍAS	1.10%	29036922
SERVIDORES	0.02%	454120
<b>Total general</b>	<b>100.00%</b>	<b>2643562515</b>

**Tabla 24:** Total de bytes del Proxy Panda a la Red UPB el 20 de Mayo de 2006.  
Creada por el autor.





**Figura 51:** Total de bytes de Proxy Panda a la Red UPB el 20 de Mayo de 2006.  
Creada por el autor.

### 3.3.7 Análisis de los resultados obtenidos de la utilización del canal de Internet.

De los resultados obtenidos el 18 de Mayo de 2006, se puede concluir que, el total de bytes intercambiado del Proxy Panda a la Red UPB supera en un factor de 5.2 al total de bytes intercambiado del la Red UPB al Proxy Panda, aunque la diferencia de archivos trasmitidos es de solo 628.

De los resultados obtenidos el 19 de Mayo de 2006, se puede concluir que, el total de bytes intercambiado del Proxy Panda a la Red UPB supera en un factor de 6.8 al total de bytes intercambiado del la Red UPB al Proxy Panda, aunque la diferencia de archivos trasmitidos es de solo 1478.

De los resultados obtenidos el 20 de Mayo de 2006, se puede concluir que, el total de bytes intercambiado del Proxy Panda a la Red UPB supera en un factor de 4.6 al total de bytes intercambiado del la Red UPB al Proxy Panda, aunque la diferencia de archivos trasmitidos es de solo 221.

Finalmente, de los resultados obtenidos el 18, 19 y 20 de Mayo de 2006, se puede concluir que, el uso del ancho de banda del canal es asimétrico, con mayor necesidad de ancho de banda de Internet hacia la red, que desde la red hacia Internet. Se evidencia por el tamaño de los objetos trasmitidos que el comportamiento del tráfico Web con peticiones o solicitudes es hacia Internet y la trasmisión de objetos o archivos es hacia la red, en consecuencia, el enlace se utiliza de manera asimétrica; y como el canal Telecom, es un canal dedicado con la misma capacidad de ancho de banda de la red hacia Internet y viceversa, (como se estableció en la sección 3.2) solo es necesario dimensionar el ancho de banda de Internet hacia la red, debido a que es mucho mayor que el de la red hacia Internet.

### **3.3.8 Comportamiento del tráfico Web.**

Según [40] el comportamiento del trafico Web es autosimilar y la distribución típica que describe el tamaño de los archivos (en el presente trabajo objetos de una página Web) es la distribución de Pareto. Para comprobarlo, se seleccionaron los archivos del día 19 de Mayo del Proxy Panda hacia la Red UPB y se catalogaron como se muestra en la tabla 25:

<b>EVENTO</b>	<b>DESCRIPCIÓN TAMAÑO DE ARCHIVO EN BYTES</b>	<b>FRECUENCIA</b>
A	0 - 10Kb	146070
B	10Kb - 100Kb	53900
C	100Kb - 1Mb	6938
D	1Mb - 10Mb	583
E	10Mb - 100Mb	16
<b>TOTAL GENERAL</b>		<b>207507</b>

**Tabla 25:** Clasificación de Eventos de acuerdo al tamaño de archivo.  
Creada por el autor.

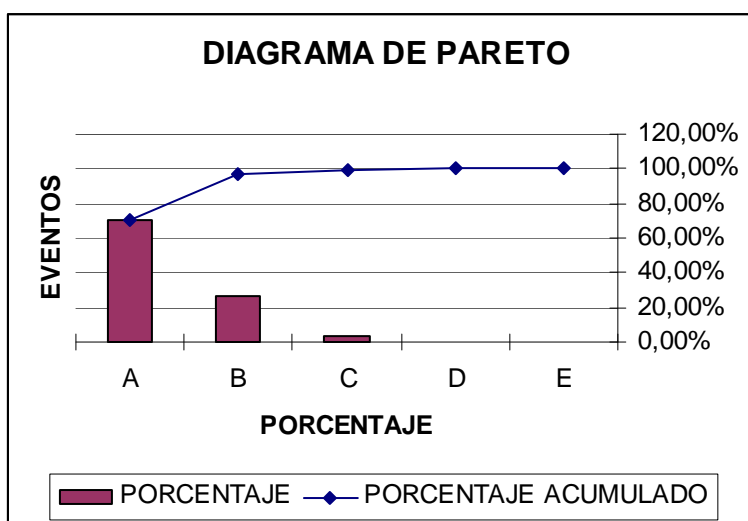
En la tabla 25, también es posible apreciar la frecuencia y la descripción de cada evento, es decir que en la red, la cantidad de archivos con un tamaño entre 10Mb y 100 Mb es tan solo de 16, a esto se le llamó evento F.

El concepto de la gráfica de Pareto fue desarrollado por el economista italiano Wilfredo Pareto, cuyo postulado indica que sólo unos pocos factores son la causa de muchos problemas. Esto es frecuentemente denominado la regla 80-20 o ley de Pareto según [19]. En teoría de redes, si el tamaño de objetos cumple con la regla 80-20, este denota el comportamiento autosimilar [40]. Esto debido a que en los parámetros del modelo CQN (ver sección 2.6.5) se establecen las aproximaciones de tráfico Web con características autosimilares.

Para aplicar un gráfico de Pareto (ver figura 52), es necesario saber la frecuencia o cantidad de archivos para cada evento determinado y su porcentaje; a continuación utilizando Excel como herramienta estadística se ordenan los eventos de acuerdo al valor de su frecuencia en orden descendente, y se determina el porcentaje acumulado como se registra en la tabla 26.

EVENTO	FRECUENCIA	PORCENTAJE	PORCENTAJE ACUMULADO
A	146070	70,39%	70,39%
B	53900	25,98%	96,37%
C	6938	3,34%	99,71%
D	583	0,28%	99,99%
E	16	0,01%	100,00%

**Tabla 26:** Frecuencia, Porcentaje y Porcentaje Acumulado de Evento.  
Creada por el autor.



**Figura 52:** Diagrama de Pareto del comportamiento del tamaño de los archivos.  
Del Proxy Panda a la Red UPB el 19 de Mayo de 2006.

Para calcular el índice de Pareto se toma la ecuación de [19] tal que:

$$\alpha = \frac{\log\left(1 + \left|\frac{f(X_{n-1})}{f(X_n)}\right|\right)}{\log\left(\frac{X_{n-1}}{X_n}\right)}$$

**Ecuación 29**

Donde:

$f(X_{n-1})$ : frecuencia del extremo inferior del último intervalo (igual a 16).

$f(X_n)$ : frecuencia del extremo superior del último intervalo (igual a 583).

$X_{n-1}$ : Extremo inferior del último intervalo (igual a 10Mb).

$X_n$ : Extremo superior del último intervalo (igual a 100Mb).

De lo anterior se obtiene que el índice de Pareto  $\alpha = 1.5$ . El cual indica el grado de autosimilitud, ya que este se encuentra relacionado con el parámetro de Hurst [40]:

$$H = \frac{3 - \alpha}{2}$$

**Ecuación 30**

Donde  $H = 0.75$ . De lo anterior se puede concluir que, el 80% de los tamaños de archivos son menores de 10Kb (ver figura 52), el 20% de los tamaños de archivos son mayores de 10Kb (ver figura 52), el tráfico Web de la UPB presenta un comportamiento de acuerdo a la ley de Pareto, donde el índice de Pareto es igual a 1.5 y el parámetro de Hurst es igual 0.75, lo cual comprueba el comportamiento autosimilar [40] del tráfico Web de la UPB seccional Bucaramanga.

## **3.4 SIMULACIÓN DEL COMPORTAMIENTO DE LA RED**

### **3.4.1 Generalidades**

Excepto para algunos casos ideales, normalmente demasiado simplistas, es extremadamente difícil evaluar las prestaciones de una red de comunicaciones compleja usando únicamente técnicas analíticas. El recurso final es siempre la experimentación directa sobre la propia red, o mejor, por razones de conveniencia práctica y económica, con algún modelo de la misma fácilmente manipulable, típicamente un programa de computadora [16]. La experimentación con este modelo es lo que se denomina simulación. La gran ventaja de la simulación es que permite reproducir fácilmente y con tanto grado de detalle como se quiera el funcionamiento de una red, por lo que el proceso de planteamiento del problema no requiere ningún tipo de abstracción matemática sofisticada. Pero esta sencillez es aparente, porque una simulación es simplemente un experimento estadístico de muestreo, que plantea un problema distinto, pero no menor: el análisis estadístico de muestras [16].

Además, la ejecución de un programa de simulación precisa de un tiempo que es directamente proporcional a la minuciosidad de la reproducción del comportamiento de la red y que puede resultar inaceptable, lo que obligaría bien a realizar abstracciones de ciertos detalles irrelevantes de la red o a sustituirlos por fórmulas matemáticas (modelado híbrido), bien a emplear técnicas estadísticas para la minimización del tamaño de las muestras o bien a repartir la carga del procesos de la simulación entre varios procesadores [16].

### 3.4.2 Tipos de Modelos de Simulación

Los modelos de simulación se pueden clasificar atendiendo diferentes criterios según [18]:

De acuerdo al instante temporal que representan:

- Modelos Estáticos: representan a un sistema en un instante determinado.
- Modelos Dinámicos: representan a un sistema que evoluciona a lo largo del tiempo.

Dependiendo de la aleatoriedad de sus variables de estado:

- Modelos Deterministas: la representación del sistema no contiene ninguna variable de estado aleatoria.
- Modelos Estocásticos o aleatorios: la representación del sistema contiene al menos una variable de estado no determinista.

Teniendo en cuenta el modo en que evolucionan sus variables de estado:

- Modelos Discretos o de eventos discretos: si las variables de estado del modelo varían en un conjunto contable de instantes de tiempo.
- Modelos Continuos: si las variables de estado varían de modo continuo en función del tiempo.

Por tanto, en este trabajo se identifica claramente que el modelo de simulación usado es dinámico, estocástico y discreto.

Es importante tener en cuenta que en la sección 2.1, se describió que principalmente se aplican las soluciones mediante simulación debido a que no puede deducirse un sistema de ecuaciones o porque la aplicación de algoritmos es demasiado compleja, otras de las ventajas y desventajas señaladas por [16, 18, 22] para los modelos de simulación, se listan a continuación:

### **Ventajas**

- Permite estudiar el comportamiento de un sistema bajo un conjunto de condiciones de operación predeterminadas.
- Una vez construido, el modelo puede ser modificado de manera rápida con el fin de analizar diferentes políticas o escenarios.
- Generalmente es más barato mejorar el sistema vía simulación, que hacerlo directamente en el sistema real.
- Es mucho más sencillo comprender y visualizar los métodos de simulación que los métodos puramente analíticos.
- Los métodos analíticos se desarrollan casi siempre, para sistemas relativamente sencillos donde suele hacerse un gran número de suposiciones o simplificaciones, mientras que con los modelos de simulación es posible analizar sistemas de mayor complejidad o con mayor detalle.
- En algunos casos, la simulación es el único medio para lograr una solución.



## **Desventajas**

- Los modelos de simulación en una computadora son costosos y requieren mucho tiempo para desarrollarse y validarse.
- El sistema se comportará de forma diferente en cada ejecución, ya que tenemos un modelo estocástico.
- Se requiere gran cantidad de corridas computacionales para encontrar soluciones “óptimas”, lo cual repercute en otros costos.
- Es difícil aceptar los modelos de simulación y siempre es necesario validar el modelo de simulación.
- Los modelos de simulación no dan soluciones óptimas.
- La solución de un modelo de simulación puede dar al analista un falso sentido de seguridad.
- La elevada carga computacional que requiere un modelo dinámico, estocástico y discreto hace imprescindible el uso del computador como herramienta para su implementación.

### **3.4.3 Proceso de desarrollo de un modelo de simulación**

La metodología para la creación y desarrollo de una simulación se puede resumir en los siguientes pasos según [16]:

- **Definición del sistema:** Cada estudio debe comenzar con una descripción de problema o del sistema. Si la descripción es dada por los tomadores de decisiones, el analista debe asegurarse que se encuentre completa. Es decir, que exista una correcta identificación del objetivo, de las variables de decisión, las restricciones, la medida de efectividad y las variables no controlables y su comportamiento estadístico.
- **Análisis del sistema:** Deben descubrirse las interacciones lógicas entre las variables de decisión, de tal suerte que se optimice la medida de efectividad en función de las variables no controlables, sin olvidar las restricciones del sistema. Con el fin de analizar un sistema, es indispensable definir algunos términos.
  - **El estado:** es el conjunto de variables que definen al sistema en cualquier instante.
  - **Un evento:** representa un acontecimiento instantáneo que modifica el estado del sistema.
  - **Actividad:** representa el tiempo requerido para llevar a cabo una operación.
  - **Entidad:** es cualquier objeto dentro del sistema, esta entidad puede ser estática o dinámica, en este último caso se denota como una *transacción* y su principal característica es su movimiento a través de las entidades estáticas del sistema. Las entidades contienen propiedades llamadas atributos que permiten crear diferencias entre ellas.

- **Formulación del modelo:** Consiste en generar un código lógico matemático que defina en forma exacta las interacciones entre las variables; debe ser una definición sencilla pero completa del sistema. Al generar las interacciones es importante tener en cuenta que se va a llevar a cabo a través del tiempo y que el uso de listas o cadenas de eventos darán la pauta en el manejo de las variables.

Una lista es un arreglo en el que se van ordenando las transacciones de acuerdo con la secuencia de eventos en el tiempo. Existen dos tipos de listas, las llamadas de eventos futuros donde la secuencia depende del tiempo de ocurrencia del evento, y las llamadas de eventos actuales cuya secuenciación depende de la ocurrencia de otro evento.

- **Selección del lenguaje:** De la selección del lenguaje dependerá el tiempo de desarrollo del modelo de simulación, es importante utilizar el lenguaje que mejor se adapte a las necesidades de simulación que se requieran. La selección puede ser desde usar un lenguaje general como lo es BASIC, PASCAL o FORTRAN hasta hacer uso de un paquete específicamente desarrollado para simular sistemas de redes; un ejemplo de este último tipo es NS, un simulador orientado a objetos escrito en C++, con un interprete de Otcl como interfaz para comandos y configuración de redes y protocolos según [42].

- **Codificación del modelo.** Consiste en generar las instrucciones o código computacional necesario para lograr que el modelo pueda ser ejecutado en algún tipo de computadora. La duración de este proceso está directamente relacionado con la selección del lenguaje.

- **Validación del modelo:** Es el proceso que tiene como objetivo determinar el grado de cercanía que tiene un modelo para representar la realidad. La validación se lleva a cabo mediante la comparación estadística entre los resultados del modelo y los resultados reales.
- **Experimentación:** En este paso se determinan las diversas alternativas que pueden ser evaluadas, seleccionando las variables de entrada y sus diferentes niveles con la finalidad de optimizar las variables de respuesta del sistema real.
- **Implantación:** Una vez seleccionada la mejor alternativa, es importante llevarla a la práctica; en muchas ocasiones este último paso es el más difícil ya que se tiene que convencer a la alta dirección y al personal de las ventajas de esta puesta en marcha. Para esto se recomienda llevar a cabo un proceso de socialización que permita visualizar el comportamiento de las variables en el sistema. Al implantar, hay que tener cuidado con las diferencias que pueda haber con los resultados simulados, ya que estos últimos se obtienen, de un modelo representativo a partir de algunas suposiciones.
- **Monitoreo y control:** No hay que olvidar que los sistemas son dinámicos y con el transcurso del tiempo es necesario modificar el modelo de simulación ante los nuevos cambios del sistema real, con el fin de llevar a cabo actualizaciones periódicas que permitan que el modelo siga siendo una representación del sistema.

Se hace necesario resaltar que el objetivo del presente trabajo no incluye el desarrollo de la implantación ni el Monitoreo y control del modelo de simulación, y se relacionan de manera solamente informativa.

#### 3.4.4 Descripción del Modelo de Red en Ns-2

Toda simulación en NS-2 comienza con el siguiente comando:

```
set ns [new Simulator]
```

Esta línea declara una nueva variable ns usando el comando *set*, para obtener archivos de salida con datos de la simulación (trace files) o archivos para visualización (nam y Xgraph files) [37].

A continuación, dentro del script generalmente se encuentra la descripción del autor sobre las características del sistema a simular, es necesario resaltar que estos y todos los comentarios se encuentran precedidos del carácter # según [37]:

```
# Aquí se simulan numerosas Fuentes de tráfico TCP que comparten un  
# Cuello de botella o un solo destino como el canal Telecom de la red UPB  
# Su número esta dado por el parámetro NodeNb  
# Para simular la red de la UPB seccional Bucaramanga en Mayo de 2006  
# Se fija el valor de la variable NodeNb en 235  
# S(1) ----  
# . |  
# . ---- N ----- D(1)  
# . |  
# S(NodeNb) ----
```

A continuación, se crea el archivo que contendrá el tiempo de traslado de las diferentes conexiones:

```
set Out [open Out.ns w]
```

Luego, se crea el archivo que contendrá el número de conexiones:

```
set Conn [open Conn.tr w]
```

Seguidamente, se abren los archivos que contendrán los datos de la simulación:

```
set tf [open out.tr w]  
$ns trace-all $tf
```

Se definen además tres archivos adicionales que serán usados para trazar, el tamaño de la cola, el ancho de banda y las pérdidas en el cuello de botella:

```
set qsize [open tamañocola.tr w]  
set qbw [open anchodebanda.tr w]  
set qlost [open perdidast.tr w]
```

Luego, se establece la topología de la red. Como deben crearse muchas conexiones, es muy útil en vez de definir cada nodo, enlace, conexión y aplicación de manera individual, definirlos como vectores usando el lenguaje Tcl [37]. Para ello, se definen los nodos

```
set N [$ns node]  
set D [$ns node]
```

Se define el enlace que conectará los nodos, lo que significa que los nodos N y D estarán conectados usando un enlace bidireccional que tiene un retardo de propagación de 1ms y

una capacidad de 2 MB en cada dirección, al igual que el canal de Telecom de la Red UPB seccional Bucaramanga:

```
$ns duplex-link $N $D 2Mb 1ms DropTail
```

Adicionalmente, se define el límite de la cola del enlace.

```
$ns queue-limit $N $D 3000
```

Se asigna el valor del número de las fuentes en este caso 235 (ver sección 3.2).

```
set NodeNb 235
```

También se asigna el valor del número de flujo por fuente:

```
set NumberFlows 830
```

Ahora se crean los usuarios (los S(j))y sus enlaces con el Proxy Panda, al cual se llama N:

```
for {set j 1} {$j<=$NodeNb} {incr j} {  
set S($j) [$ns node]  
$ns duplex-link $S($j) $N 100Mb 1ms DropTail  
$ns queue-limit $S($j) $N 1000  
}
```

Después de establecer la topología de la red, es necesario crear el tráfico, definir la ruta de ese tráfico, y los protocolos del mismo como se encuentra descrito en [37]. Para ello, se crean las fuentes (tcpsrc) y destinos (tcp\_snk) TCP [37]:

```

for {set i 1} {$i<=$NodeNb} {incr i} {
for {set j 1} {$j<=$NumberFlows} {incr j} {
set tcpsrc($i,$j) [new Agent/TCP]
set tcp_snk($i,$j) [new Agent/TCPSink]
}
}

```

Luego, se crean las conexiones entre los clientes TCP y el servidor Panda [37]:

```

for {set i 1} {$i<=$NodeNb} {incr i} {
for {set j 1} {$j<=$NumberFlows} {incr j} {
$ns attach-agent $S($i) $tcpsrc($i,$j)
$ns attach-agent $D $tcp_snk($i,$j)
$ns connect $tcpsrc($i,$j) $tcp_snk($i,$j)
}
}

```

Después, se crean las fuentes FTP, es importante aclarar que el tráfico de la UPB es HTTP. Sin embargo, en su lugar se usará el protocolo FTP, debido a que se desea simular el comportamiento TCP para validar el modelo de red de Cola Cerrada CQN y este modela el comportamiento de TCP y no el de HTTP específicamente, por lo cual se toma la única opción disponible en NS-2. Aunque por tratarse al nivel de archivos (objetos) las diferencias no son significativas.

```

for {set i 1} {$i<=$NodeNb} {incr i} {
for {set j 1} {$j<=$NumberFlows} {incr j} {
set ftp($i,$j) [$tcpsrc($i,$j) attach-source FTP]
}
}

```

Como se describió en la primera parte de este capítulo, el modelo de simulación es estocástico, para lo cual al menos una variable debe ser aleatoria. En consecuencia se



generará de manera aleatoria el tamaño de los archivos, usando la distribución Pareto y los tiempos de llegada con la distribución exponencial según [17], por lo anterior se tiene que el comportamiento del tráfico Web cumple con la Distribución de Pareto como se describió en la sección 3.3.8. Para esto se establecen las dos variables aleatorias mediante el comando *set*:

```
set rng1 [new RNG]  
$rng1 seed 0  
set rng2 [new RNG]  
$rng2 seed 0
```

A continuación, se definen las características de las distribuciones de las variables:

- Para la variable con distribución exponencial se tiene como valor medio, aquel obtenido de la caracterización del tráfico y que corresponde a la tasa del nodo PS ( $\mu_2$ , avg)

```
set RV [new RandomVariable/Exponential]  
$RV set avg_ 0.135  
$RV use-rng $rng1
```

- Para la variable con distribución de Pareto tiene como valor medio, aquel obtenido de la caracterización del tráfico y que corresponde al tamaño promedio de archivos ( $f$ , avg). De la misma manera se establece el índice de Pareto (shape) obtenido en la sección 3.3.8.

```
set RVSize [new RandomVariable/Pareto]  
$RVSize set avg_ 1467.21  
$RVSize set shape_ 1.5  
$RVSize use-rng $rng2
```

También, se hace necesario definir el inicio y el tamaño de las conexiones, es importante resaltar el proceso de llegadas es Poisson según [17, 37], para esto se establece:

```

for {set i 1} {$i<=$NodeNb} { incr i } {
  set t [$ns now]
  for {set j 1} {$j<=$NumberFlows} { incr j } {

```

El tiempo de inicio de transferencia para cada fuente i, usando la variable aleatoria de distribución exponencial:

```

      set t [expr $t + [$RV value]]
    set Conct($i,$j) $t

```

El tamaño de la siguiente transferencia desde la fuente i, usando la variable aleatoria con distribución de Pareto:

```

      set Size($i,$j) [expr [$RVSize value]]
    $ns at $Conct($i,$j) "$ftp($i,$j) send $Size($i,$j)"

```

Actualización del número de flujos

```

      $ns at $Conct($i,$j) "countFlows $i 1"
    }
  }

```

En el punto del modelo de simulación donde todos los parámetros se encuentran definidos, se establece el proceso de finalización (proc finish), donde además se cierran los archivos creados y abiertos previamente. También se definen los comandos para obtener los resultados gráficos del modelo de simulación, tales como el tamaño de la cola, el ancho de

banda y el número de paquetes perdidos en el tiempo de simulación. Para esto se usa Xgraph, una herramienta de dibujo o trazado que esta incluida en el paquete de NS a la que debe describirse detalladamente el titulo y demás propiedades que desea visualizar [37] .

```
proc finish {} {
    global ns tf qsize qbw qlost
    $ns flush-trace
    close $qsize
    close $qbw
    close $qlost

    exec Xgraph tamañodecola.tr -geometry 800x400 -t "Tamaño de Cola" -x "secs"
    -y "# packets" &
    exec Xgraph anchodebanda.tr -geometry 800x400 -t "Ancho de Banda" -x "secs"
    -y "Kbps" -fg white &
    exec Xgraph pérdidas.tr -geometry 800x400 -t "# Paquetes Perdidos" -x "secs" -y
    "packets" -fg red &
    exit 0
}
```

Antes de finalizar, es necesario establecer el procedimiento que monitorea la cola y determinar la información que debe ir en los archivos creados anteriormente para que se puedan visualizar las variables de interés de manera gráfica al final de la simulación.

```
set qfile [$ns monitor-queue $N $D [open queue.tr w] 0.05]
[$ns link $N $D] queue-sample-timeout;

proc record {} {
    global ns qfile qsize qbw qlost N D
    set time 0.05
    set now [$ns now]

    # print in the file $qsize el tamaño de cola
    # print in the file $qbw el ancho de banda usado
    # print in the file $qlost la tasa de pérdidas
    $qfile instvar parrivals_ pdepartures_ bdrops_ bdepartures_ pdrops_
    puts $qsize "$now [expr $parrivals_-$pdepartures_-$pdrops_]"
    puts $qbw "$now [expr $bdepartures_*8/1024/$time]"
    set bdepartures_ 0
    puts $qlost "$now [expr $pdrops_/$time]"
    $ns at [expr $now+$time] "record"
}
```

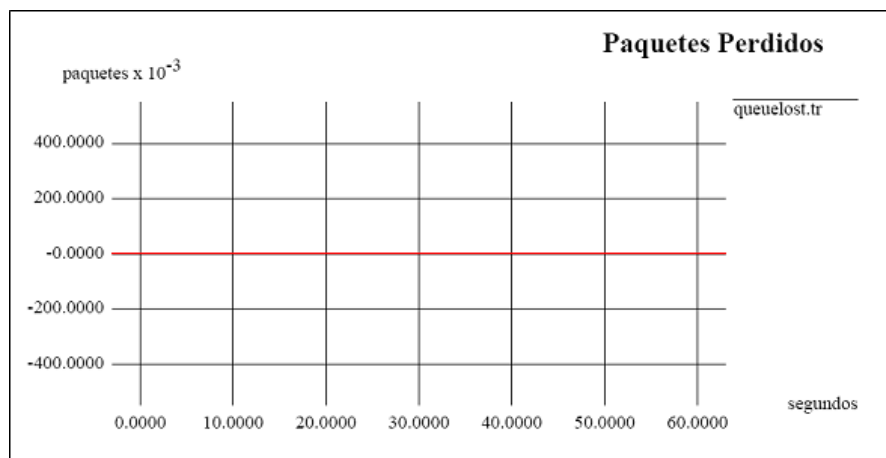
Para finalizar se establecen los tiempos de inicio y fin de la simulación, y el comando que da inicio a la misma.

```
$ns at 0.0 "record"  
$ns at 20 "finish"  
  
$ns run
```

### 3.4.5 Resultados obtenidos de la simulación

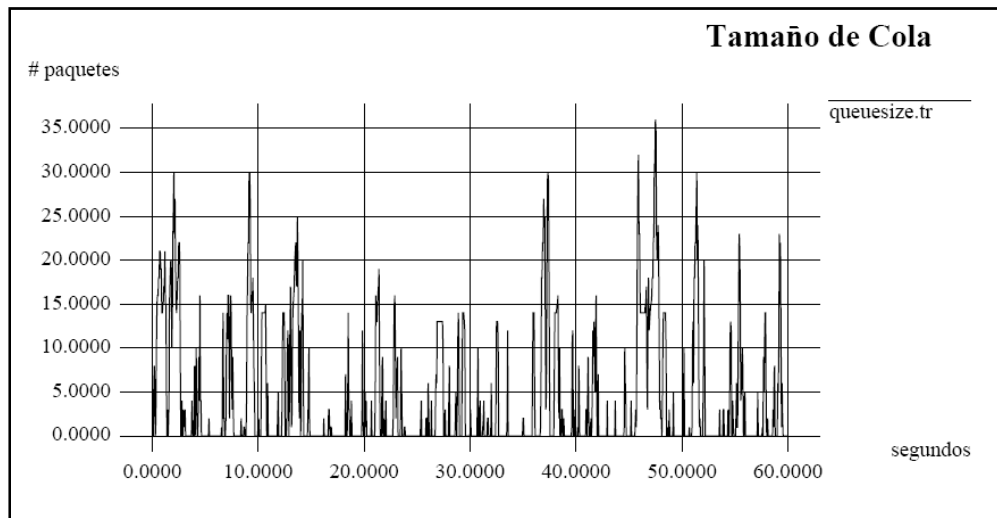
Teniendo en cuenta el código de simulación relacionado en la sección anterior, usando como parámetros el índice de Pareto (ver sección 3.3.8) de 1.5, el tamaño promedio de archivo de 1467.21 bits, y el tiempo entre conexiones de 0.135 segundos.

Como se encuentra implementado el control de congestiónamiento de TCP se espera la pérdida mínima de paquetes, en este caso, no hubo pérdidas como se observa en la figura 53. Esta característica cumple con las condiciones del modelo CQN (ver sección 2.6).



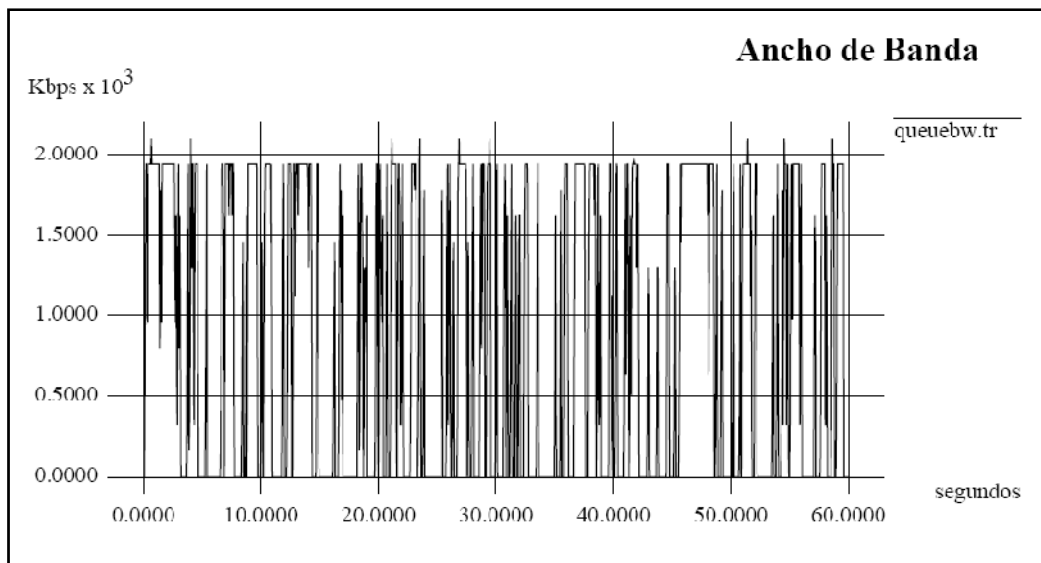
**Figura 53:** Pérdida de paquetes en el enlace que representa el cuello de botella.

Como se encuentra implementado el control de congestión de TCP, se verifica que hay paquetes en la cola, lo que está acorde con las condiciones del modelo CQN (ver sección 2.6). Esta variación de la cola se observa en la figura 54.



**Figura 54:** Número de paquetes en el enlace que representa el cuello de botella.

Finalmente se obtiene la gráfica que representa el consumo de ancho de banda sobre el enlace que representa el cuello de botella. En la figura 55.



**Figura 55:** Ancho de banda usado sobre el enlace representa el cuello de botella.

Esta gráfica representa, la variación del ancho de banda usado sobre el enlace, a través del tiempo, donde se obtiene que en el peor caso de congestión, el ancho de banda usado del canal es igual a 2100Kbps.

### 3.5 APLICACIÓN DEL MODELO DE RED DE COLA CERRADA

En el capítulo 3, se definió el modelo de red de cola cerrada para estudiar sistemas de control de flujo como el de TCP. En el caso general el modelo de cola cerrada incluye realimentación de paquetes, ya que en las redes de cola cerrada el número total de paquetes se mantiene constante. En el modelo se considera que el arribo es un proceso de Poisson homogéneo, la distribución de probabilidad del tiempo de servicio es exponencial y la

longitud de tamaño de archivo tiene distribución de Pareto. Bajo las condiciones anteriores la red de colas se modela como forma-producto, del tipo Jackson, donde la tasa de arribo depende del número total de archivos en la red y el tiempo de servicio depende del estado se su respectiva cola [17]. Esta clase de ecuaciones puede ser derivada de la media del número de tareas en las estaciones de servicio, como se observa en la siguiente ecuación [3]:

$$\mathbf{B} = N * \mathbf{h}$$

**Ecuación 27**

Donde:

- El número de Fuentes  $N$ .
- La capacidad de canal garantizado a cada usuario denotada por  $\mathbf{b}$ .
- La media de servicio en el nodo **PS** denotada por  $\mu_2^{-1}$ .
- La media de tiempo en el nodo **IS** denotado por  $\mu_1^{-1}$ .
- La media de tamaño de archivo denotado por  $f$ .
- La capacidad del enlace seleccionado denotada por  $\mathbf{B}$ .

- $$h = \left( \frac{1}{b} + \frac{1}{\mu_1 f} \right)^{-1}$$

**Ecuación 28**

Para estimar la capacidad del enlace de Telecom de la red UPB seccional Bucaramanga, fue necesario determinar a partir de los datos capturados en Mayo de 2006, los parámetros anteriormente descritos, para evaluar el modelo de red de cola cerrada.

El procesamiento de datos para la determinación de los parámetros necesarios para evaluar el modelo de red de cola cerrada solo debe hacerse sobre la hora de mayor tráfico, del día de mayor tráfico, por tanto se seleccionó el jueves 18 de Mayo de 2006 de las 11:49:40 a.m a las 12:50:60 p.m. de acuerdo a la información suministrada en el capítulo 5.

- El número de Fuentes  $N$  es igual a 235 y fue determinado a partir de la información dada por el administrador de la red tal como se muestra en la Sección 3.2.
- La capacidad de canal garantizado  $b$  se determina a partir del rango  $10^4$ - $10^6$  bps y se establece en 50kbps.

### **3.5.1 Construcción del algoritmo en Excel.**

Obtener los siguientes parámetros exige el procesamiento de los datos demostrado en la sección 3.3 (solo con los archivos correspondientes a la hora de máximo tráfico). Además de la aplicación del código de programación en anexo 1 y anexo 2 sobre la herramienta estadística Excel:

- El código toma el archivo de texto generado por Sed “*resultadoso.txt*” y lo convierte en una hoja de calculo “*resultadoso.xls*” ó el archivo “*resultadosi.txt*” para convertirlo en la hoja de calculo “*resultadosi.xls*”.



- Inserta nuevas columnas para calcular y establecer el tiempo de inicio de conexión, el tiempo fin de conexión y el tiempo que duró la conexión así:
  - $\text{Tiempo i} = (D2*60*60)+(E2*60)+F2$
  - $\text{tiempo f} = (K2*60*60)+(L2*60)+M2$
  - $\text{tiempo on} = P2-O2$
- En el caso del archivo resultadoso.xls, se crea una nueva columna con el fin de escribir de manera adecuada la IP destino así:
  - $=\text{CONCATENAR}(W2,".",X2,".",Y2,".",Z2)$
- Finalmente se obtiene una hoja de calculo con la siguiente información:

semanai	mesi	diai	horai	minutosi	segundosi	anoi
Thu	5	18	10	18	10	2006

**Tabla 27:** Muestra de la fecha y hora de inicio de conexión.

semanao	meso	dio	horao	minutoso	segundoso	anoo
Thu	5	18	10	18	10	2006

**Tabla 28:** Muestra de la fecha y hora del fin de conexión.

TIEMPO I	TIEMPO F	TIEMPO ON
37090	37090	0

**Tabla 29:** Muestra del tiempo inicio, tiempo fin y tiempo que duró la conexión en segundos.

ips1	ips2	ips3	ips4	IPS	puertos
172	16	6	5	172.16.6.5	1086

**Tabla 30:** Muestra de la IP y puerto de origen de la conexión obtenida de el archivo resultadoso.xls.

ips1	ips2	ips3	ips4	puertos
172	16	12	4	53273

**Tabla 31:** Muestra de la IP y puerto de origen de la conexión obtenida de el archivo resultadosi.xls.

ipd1	ipd2	ipd3	ipd4	puertod
172	16	12	4	50989

**Tabla 6.6:** Muestra de la IP y puerto de destino de la conexión obtenida de el archivo resultados.xls.

ipd1	ipd2	ipd3	ipd4	IPD	puertod
172	16	6	5	172.16.6.5	55805

**Tabla 32:** Muestra de la IP y puerto de destino de la conexión obtenida de el archivo resultados.xls.

cantidad	bytes
99	137794

**Tabla 33:** Muestra de la cantidad de paquetes necesarios para transmitir el archivo y el tamaño de archivo.

- La media de tiempo en el nodo **IS** denotada por  $\mu_1^{-1}$ , se calcula usando la función promedio sobre los datos de la columna Tiempo I e invirtiéndola para obtener  $\mu_1$ .
  - El valor obtenido para estimar la capacidad del enlace de Internet hacia la red es igual a 0.135 segundos.
  - El valor obtenido para estimar la capacidad del enlace de la red hacia Internet es igual a 0.135 segundos.
  
- La media de tamaño de archivo denotado por  $f$ , se calcula usando la función promedio sobre los datos de la columna bytes multiplicado por 8 para obtener bits.
  - El valor obtenido para estimar la capacidad del enlace de Internet a la red es igual a 1467.21 bits.

- El valor obtenido para estimar la capacidad del enlace de la red hacia Internet es igual a 134.22 bits.
- Se calcula el parámetro  $h$  usando la ecuación 28 ,con los valores relacionados obteniendo:
  - EL valor de  $h$  para estimar la capacidad del enlace de Internet hacia la red es igual a 8900.78.
  - El valor de  $h$  para estimar la capacidad del enlace de la red hacia Internet es igual a 974.74.
- Finalmente se ingresan los valores de los parámetros en la ecuación 27, para obtener lo siguiente:
  - El valor estimado de ancho de banda para el enlace Telecom de Internet hacia la Red UPB seccional Bucaramanga es de 2091684,95 bps. (ver figura 56)
  - El valor estimado de ancho de banda para el enlace Telecom de la Red UPB seccional Bucaramanga hacia Internet es de 229065,92 bps. (ver figura 57)

### **3.5.2 Resultados obtenidos para la hora Pico**

Las siguientes gráficas se obtienen al variar el parámetro  $b$  de 10kbps a 100kbps.

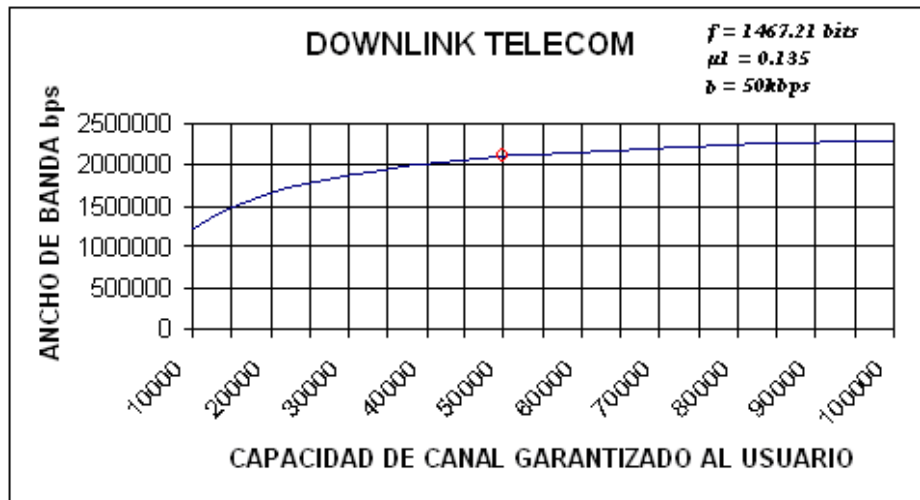


Figura 56: Capacidad necesaria en el Enlace Downlink de Telecom respecto a la capacidad de usuario.

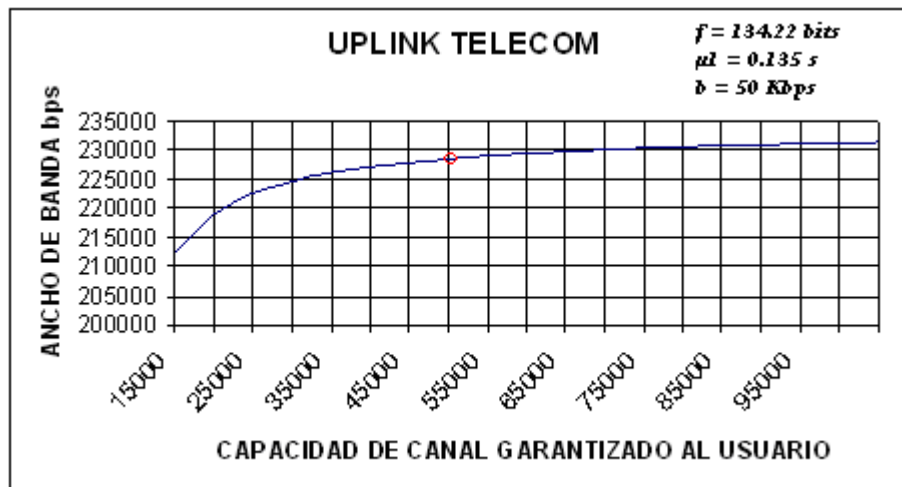
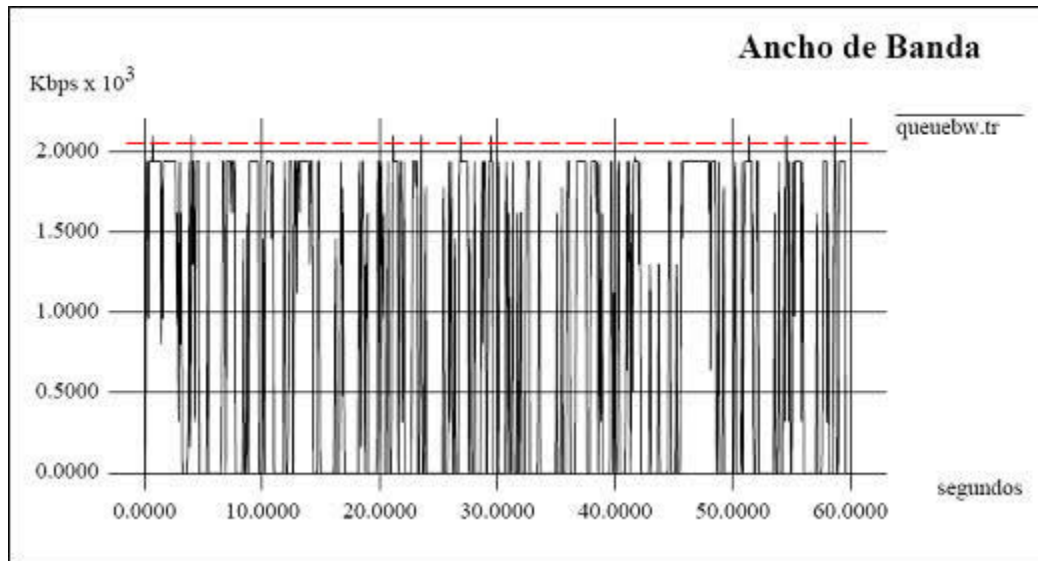


Figura 57: Capacidad necesaria en el Enlace Uplink de Telecom respecto a la capacidad de usuario.

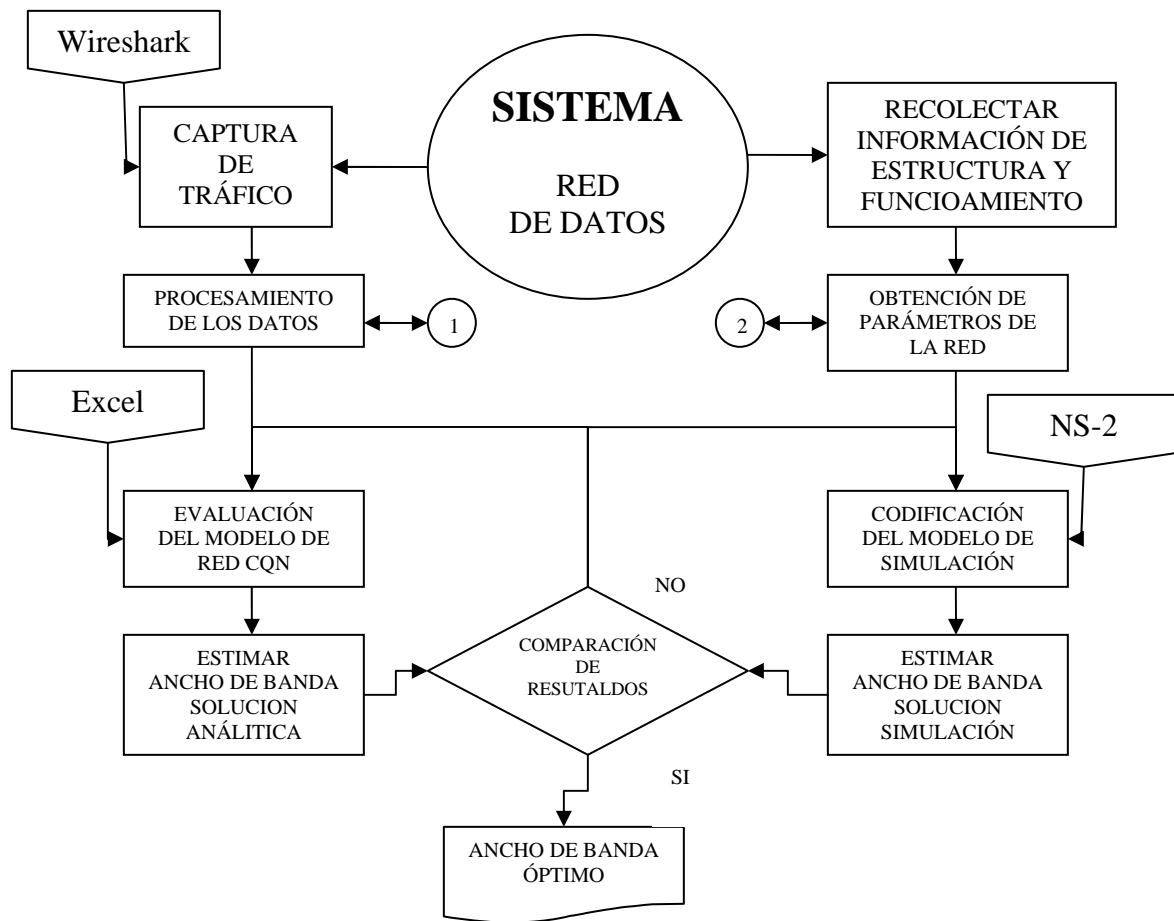
### 3.5.3 Validación del Modelo



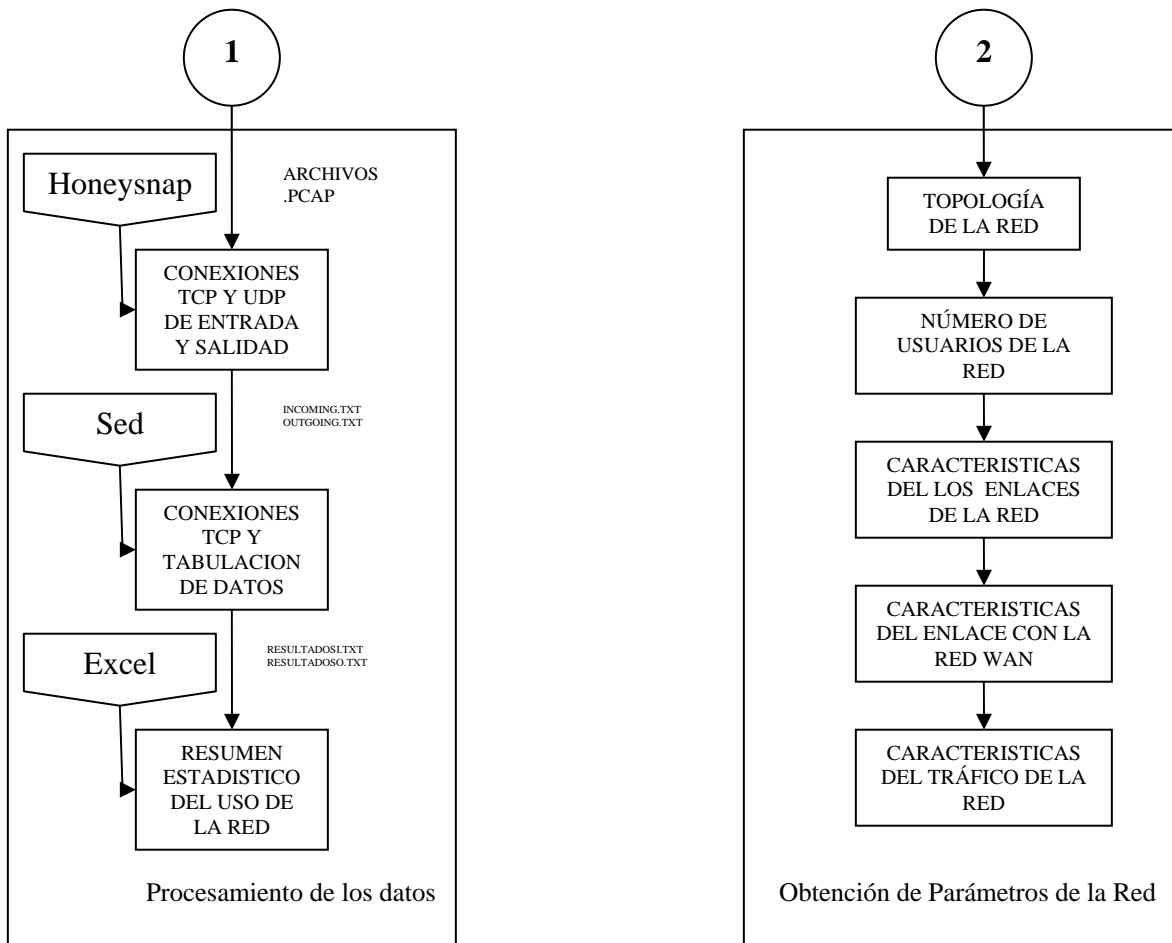
**Figura 58:** Ancho de banda usado sobre el enlace representa el cuello de botella y Capacidad necesaria en el Enlace Uplink de Telecom respecto a la capacidad de usuario.

Sobre la figura 57 que representa el consumo de ancho de banda sobre el enlace que representa el cuello de botella de la simulación (ver sección 3.4) se observa en línea punteada el valor del límite calculado con el modelo CQN, por lo que se concluye que este modelo, es una buena aproximación para calcular el ancho de banda del enlace, ya que el límite calculado es aproximadamente igual al valor límite obtenido con la simulación. Teniendo en cuenta que las condiciones de la simulación son iguales a las condiciones del cálculo. Por lo cual se declara que la hipótesis propuesta en la sección 3.0 es aceptable y válida para aplicarlos a un sistema de red con tráfico Web.

### 3.6 SINTESIS DE LA METODOLOGÍA USADA



**Figura 59:** Diagrama de Flujo “Metodología para la implementación del Modelo CQN”



**Figura 60:** Diagrama de Flujo “Metodología para la implementación del Modelo CQN” procesos 1 y 2.

## CONCLUSIONES

- El total de bytes intercambiado el 18 de Mayo de 2006 del Proxy Panda a la Red UPB supera en un factor de 5.2 al total de bytes intercambiado del la Red UPB al Proxy Panda, aunque la diferencia de archivos trasmitidos es de solo 628.
- El total de bytes intercambiado el 19 de Mayo de 2006 del Proxy Panda a la Red UPB supera en un factor de 6.8 al total de bytes intercambiado del la Red UPB al Proxy Panda, aunque la diferencia de archivos trasmitidos es de solo 1478.
- El total de bytes intercambiado el 20 de Mayo de 2006 del Proxy Panda a la Red UPB supera en un factor de 4.6 al total de bytes intercambiado del la Red UPB al Proxy Panda, aunque la diferencia de archivos trasmitidos es de solo 221.
- El uso del ancho de banda del canal es asimétrico, con mayor necesidad de ancho de banda de Internet hacia la red, que desde la red hacia Internet. Con peticiones o solicitudes hacia Internet y trasmisión de objetos o archivos hacia la red.
- Como el canal Telecom es un canal dedicado con la misma capacidad de ancho de banda de la red hacia Internet y viceversa, solo es necesario dimensionar el ancho de



banda de Internet hacia la red, debido a que es mucho mayor que el de la red hacia Internet.

- El tráfico Web de la UPB presenta un comportamiento de acuerdo a la ley de Pareto, con índice igual 1.5 y parámetro de Hurst equivalente de 0.75, lo que demuestra autosimilitud.
- En la Red de la UPB seccional Bucaramanga, se determino que el protocolo de aplicación presente es el HTTP 1.0, en consecuencia por cada conexión TCP se transmite un objeto de una página Web.
- Como los tamaños de los objetos de las páginas Web son tan pequeños se pudo determinar que el 80% los tamaños de archivos transmitidos desde Internet hacia la red UPB seccional Bucaramanga son menores o iguales a 10 KB.
- Como el 80% de los archivos transmitidos de Internet hacia la red UPB seccional Bucaramanga son menores o iguales 10Kb, estos serán transmitidos mientras todavía se este ejecutando el algoritmo de arranque lento de TCP.
- En las aplicaciones de datos elásticas, como el tráfico HTTP, el flujo de segmentos se adapta al ancho de banda disponible gracias a los algoritmos de control de congestión que actúan sobre TCP.

- El procesamiento de datos para la determinación de los parámetros necesarios para evaluar el modelo de red de cola cerrada solo debe hacerse sobre la hora de mayor tráfico, del día de mayor tráfico, de lo contrario los resultados obtenidos no serían confiables.
- El modelo de red de cola cerrada representa un modelo de población Finita que consta de  $N$  generadores de tareas tal que, cada uno de ellos, se comporta de la siguiente manera; tras generar una tarea, ésta accede al sistema de espera (cola y recurso) y una vez que el procesamiento de esta tarea ha terminado, el generador espera un tiempo aleatorio, modelado según la variable aleatoria que representa el tiempo de ocio, antes de generar la tarea siguiente.
- El modelo de red de cola cerrada representa el control retroalimentado sobre un enlace que representa un cuello de botella, manteniendo algunos bytes en la cola, con la mínima pérdida de paquetes posible.
- El modelo de red de cola cerrada del presente trabajo, es un modelo del tipo forma-producto, resultado de la unión retroalimentada de una cola  $M/M/1$  y una cola  $M/P/N$ , donde  $N$  representa el número de usuarios de la red.
- Tanto para el modelo de red de cola cerrada como para el modelo de simulación se considera que el arribo es un proceso de Poisson homogéneo, la distribución de

probabilidad del tiempo de servicio es exponencial y la longitud de tamaño de archivo tiene distribución de Pareto.

- La evaluación del modelo de red de cola cerrada se realiza mediante la expresión  $B = N * h$ , donde “h” representa la media armónica de dos velocidades que se encuentran naturalmente asociadas al tráfico de datos elástico dentro de los límites y condiciones de la red. Da como resultado una aproximación aceptable para el dimensionamiento del enlace congestionado.
- Aunque en la captura de datos, se tuvo en cuenta el efecto de cache, los resultados obtenidos son los del ancho de banda del enlace y no los del servidor Proxy, ya que precisamente el cache se creó para optimizar los recursos, se produciría por el contrario un error de sobre dimensionamiento de la capacidad del enlace.
- Los resultados de evaluación del modelo de red de cola cerrada son una representación de la realidad de la red en Mayo de 2006; comparándolo con los resultados del modelo de simulación. Estos resultados pueden extrapolarse, intercambiando el número de usuarios, para dimensionar el canal hoy en día.
- Como en el modelo se usó el tamaño de los objetos característicos del comportamiento HTTP 1.0, para dimensionar el canal y actualmente el tráfico HTTP es del tipo 1.1 donde en una sola conexión TCP se obtiene toda la página, se

recomienda volver a capturar el tráfico de la red, ya que el tamaño promedio de archivo y el tiempo entre conexiones varia.

- El modelo se encuentra limitado a redes donde el tráfico sea elástico, para dimensionar el cana, al igual que en el presente trabajo, de lo contrario no pueden garantizarse los resultados obtenidos.
- Para futuros trabajos podría estudiarse el comportamiento del tráfico elástico sobre redes inalámbricas, con el fin de determinar el número máximo de usuarios, a los que una red de este tipo puede ofrecer un servicio aceptable.

## BIBLIOGRAFÍA

1. Betancourt, D., *Dimensionamiento de capacidad de canal en un enlace de Internet en redes congestionadas*. 2002, Universidad de los Andes: Bogota. p. 85.
2. Gomez, A.M. and P. Ortega, *Simulación y análisis de tráfico presente en la red de datos de la universidad Pontificia Bolivariana Seccional Bucaramanga*, in *Facultad de Ingeniería Electrónica*. 2006, Universidad Pontificia Bolivariana: Bucaramanga.
3. Arthur W. Berger, Y.K., *Dimensioning Bandwidth for Elastic Traffic in High-Speed Data Networks*. IEEE/ACM Transactions on Networking (TON) 2000. **Volume 8** (Issue 5 ): p. 643 - 654
4. D. P. Heyman, T. V. Lakshman, and A.L. Neidhardt, *A new method for analyzing feedback-based protocols with applications to engineering web traffic over the internet*. ACM SIGMETRICS'97, Performance Evaluation, 1997: p. 24–38.
5. Will E. Leland, et al., *On the Self-Similar Nature of Ethernet Traffic*. ACM SIGCOMM Computer Communication Review, 1993. **Volume 23** (Issue 4): p. 11.
6. Bergeraand, A. and Y. Koganb, *Multi-Class Elastic Data Trac: Bandwidth Engineering Via Asymptotic Approximations*. IEEE/ACM Transactions on Networking, 2002: p. 10.
7. Y·UZ·UKIRMIZI, M., *Performance Evaluation of Closed Queueing Networks with Limited Capacities*. Turkish J. Eng. Env. Sci., 2006.
8. A. W. Berger1, E.G.C.a.Y.A.K., *Network Engineering of Elastic Data Traffic via Tandem Queueing Network Models*. IEEE/ACM Transactions on Networking.
9. Berger, A., L. Bregman, and Y. Kogan, *Bottleneck analysis in multiclass closed queueing networks and its application*. J.C. Baltzer AG, Science Publishers, 1999.
10. Mark E. Crovella, A.B., *Self-similarity in World Wide Web Traffic: Evidence and Possible Causes* IEEE/ACM Transactions on Networking 1997: p. 10.
11. Bonald, T. and J. Roberts, *Performance modeling of elastic traffic in overload*.
12. Benameur, N., et al., *Integrated Admission Control for Streaming and Elastic Tra±c*.
13. Arias, D.F., *Análisis De Tráfico De La Red De Datos De La Universidad Pontificia Bolivariana Bucaramanga*, in *Facultad de Ingeniería Electrónica*. 2005, Universidad Pontificia Bolivariana: Bucaramanga.

14. Monsalve, J.C.E. and J.A.M. Valero, *Estudio del Nivel de Seguridad de la red Lan de la UPB Bucaramanga*, in *Facultad de Ingenieria Electrónica*. 2006, Universidad Pontificia Bolivariana: Bucaramanga.
15. Mogollón, J.A.M., et al., *Caracterización y simulación del tráfico de redes LAN mediante el modelo MMPP*. Rev. Fac. Ing. Univ. Antioquia., 2007.
16. Azarang, M.R. and E.G. Dunna, *Simulación y análisis de modelos estocásticos*. Segunda ed, ed. M. Hill. 1996, Mexico. 281.
17. Cifuentes, R.C., *Teoría de Colas: Aplicación a las telecomunicaciones*. Primera ed, ed. U. Distrital. 1997, Santa Fe de Bogotá D.C. 312 p.
18. Arias, J.J.P., A.S. González, and R.D. Redondo, *Teoría de Colas y Simulación de Eventos Discretos*, ed. S.A. Pearson Educación. 2003, MADRID.
19. Chung., C.A., *Simulation modeling handbook : a practical approach*. INDUSTRIAL AND MANUFACTURING ENGINEERING SERIES, ed. H.R. Parsaei. 2003: CRC Press LLC. 575.
20. Held, G., *Ethernet Networks: Design, Implementation, Operation, Management*. Fourth Edition ed, ed. L. John Wiley & Sons. 2003, Georgia, USA.
21. Cisco Systems, I., *Guía del primer año. CCNA 1 y 2*. Tercera ed, ed. S.A. Pearson Educación. 2004, Madrid: John Wait. 1008.
22. Gunter, B.S., Greiner; Hermann de Meer; Kishor, Trivedi. , *Queueing Networks and Markov Chains*. Second ed, ed. J.W. Sons. 2006, New Jersey. 896.
23. Iversen, V.B., *TELETRAFFIC ENGINEERING and NETWORK PLANNING*, ed. T.U.o. Denmark. 2006.
24. Gourley, D., et al., *HTTP: The Definitive Guide*. 2002: O'Reilly. 656.
25. Shiflett, C., *HTTP Developer's Handbook* 2003: Sams Publishing. 312.
26. Naugle, M., *Illustrated TCP/IP: A Graphic Guide to the Protocol Suite*, ed. J.W. Sons. 1999, New York.
27. Jacobson, V. and M.J. Karels, *Congestion Avoidance and Control*. Proceedings of SIGCOMM '88, 1988.
28. Orebaugh, A., et al., *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. 2007: Syngress Publishing, Inc. 563.
29. Combs, G., Última consulta Mayo de 2009 en <http://www.wireshark.org/download.html>.
30. Clune, A., *USAGE for Honeysnap, with examples* Última Consulta Mayo 2009 en <https://projects.honeynet.org/honeysnap/attachment/wiki/WikiStart/USAGE>, 2008.
31. Clune, A., D. Watson, and J. Riden, Última Consulta Mayo 2009 en <https://projects.honeynet.org/honeysnap/>.
32. *Editor en línea sed (Stream Editor)* Última Consulta Mayo 2009 en <http://www.tutorialesytrucos.com/editor-en-linea-sed-stream-editor.html>.
33. Free Software Foundation, I., Última Consulta Mayo 2009 en <http://gnuwin32.sourceforge.net/packages/sed.htm>. 2007.
34. Berkeley, U., L. USC/ISI, and X. PARC., *The ns Manual (formerly ns Notes and Documentation)1*, in *The VINT Project*. 2008. p. 421.
35. Project, V., Última consulta en Mayo de 2009 en <http://www.isi.edu/nsnam/ns/ns-build.html>.
36. Roman, S., *Writing Excel Macros with VBA, 2nd Edition*. 2002: O'Reilly Media; 2nd edition. 560.

37. Altman, E. and T. Jimenez, *NS Simulator For Beginners*. Lecture Notes 2003-2004, 2003: p. 146.
38. Tanenbaum, A.S., *Redes de computadoras*. 4 ed, ed. S.A. Pearson Educación. 2003, MEXICO. 891
39. Fujinoki, H. and K.K. Gollamudi, *Object packaging - Web response time reduction for slow and busy Web servers*. LCN 2002. 27th Annual IEEE Conference on Volume , Issue , 6-8 Nov, 2002: p. 345 - 347.
40. Crovella, M.E. and A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*. In IEEE/ACM Transactions on Networking, 1997. **Vol 5**(Number 6): p. pages 835-846,.
41. B. Sikdar, S. Kalyanaraman, and K.S. Vastola, *An integrated model for the latency and steady-state throughput of TCP connections* Performance Evaluation 2001: p. 139-154.
42. Sosa, V.J.S., *ARQUITECTURA PARA LA DISTRIBUCIÓN DE DOCUMENTOS EN UN SISTEMA DISTRIBUIDO A GRAN ESCALA*, in *Programa de doctorado: ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES*. 2002, Universidad Politécnica de Cataluña: Barcelona. p. 145.
43. Cao, J., et al., *Stochastic Models for Generating Synthetic HTTP Source Traffic*. HTTP SOURCE TRAFFIC MODELING.
44. Yang, X., *Designing Trafic Proles for Bursty Internet Trafic*. MIT Laboratory for Computer Science, Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory.
45. Mao, S., S.S. Panwar, and G. Lapiotis, *The Effective Bandwidth of Markov Modulated Fluid Process Sources with a Generalized Processor Sharing Server*. New York State Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University, 2001. **0-7803-7206-9/01**: p. 2341-2346.
46. Guillemina, F., et al., *Buffer over flow bounds for multiplexed regulated traffic streams*. France Telecom, Institute for Problems of Information Transmission, Russian Academy of Sciences, School of Electrical and Computer Engineering at Purdue University.
47. Augé, J. and J. Roberts, *Buffer sizing for elastic traffic*. Next Generation Internet Design and Engineering, 2006.
48. Fajardo, G.C.G., *Metodología para la construcción de una simulación en una red de datos universitaria.*, in *Departamento de Ingeniería Eléctrica y Electrónica*. 2002, Universidad de los Andes: Bogota DC. p. 77.

## ANEXOS

**A1.** Macro para obtener archivo de texto y calcular los diferentes parámetros.

Sub Macro1()

Columns("O:O").Select

Selection.Insert Shift:=xlToRight

Columns("P:P").Select

Selection.Insert Shift:=xlToRight

Columns("Q:Q").Select

Selection.Insert Shift:=xlToRight

Range("O1").Select

ActiveCell.FormulaR1C1 = "TIEMPO I"

Range("P1").Select

ActiveCell.FormulaR1C1 = "TIEMPO F"

Range("Q1").Select

ActiveCell.FormulaR1C1 = "TIEMPO ON"

Range("Q2").Select

Columns("O:O").EntireColumn.AutoFit

Columns("P:P").EntireColumn.AutoFit

Columns("Q:Q").EntireColumn.AutoFit



```

Range("O2").Select
ActiveCell.FormulaR1C1 = "=(RC[-11]*60*60)+(RC[-10]*60)+RC[-9]"
Range("P2").Select
ActiveCell.FormulaR1C1 = "=(RC[-5]*60*60)+(RC[-4]*60)+RC[-3]"
Range("Q2").Select
ActiveCell.FormulaR1C1 = "=RC[-1]-RC[-2]"
Range("O2:Q2").Select
Selection.Copy
Range("O3:Q27354").Select
ActiveSheet.Paste
Columns("V:V").Select
Application.CutCopyMode = False
Selection.Insert Shift:=xlToRight
Range("V1").Select
ActiveCell.FormulaR1C1 = "IPS"
Range("V2").Select
ActiveCell.FormulaR1C1 = _
    "=CONCATENATE(RC[-4],""."" ,RC[-3],""."" ,RC[-2],""."" ,RC[-1])"
Range("V3").Select
Columns("V:V").EntireColumn.AutoFit
Columns("V:V").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlBottom

```

```

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Range("V2").Select

Selection.Copy

Range("V3:V27354").Select

ActiveSheet.Paste

Columns("AB:AB").Select

Application.CutCopyMode = False

Selection.Insert Shift:=xlToRight

Range("AB1").Select

ActiveCell.FormulaR1C1 = "IPD"

Range("AB2").Select

ActiveCell.FormulaR1C1 = _

    "=CONCATENATE(RC[-4],""."" ,RC[-3],""."" ,RC[-2],""."" ,RC[-1])"

Range("AB3").Select

Columns("AB:AB").EntireColumn.AutoFit

Columns("AB:AB").Select

With Selection

```

```
.HorizontalAlignment = xlCenter  
.VerticalAlignment = xlBottom  
.WrapText = False  
.Orientation = 0  
.AddIndent = False  
.IndentLevel = 0  
.ShrinkToFit = False  
.ReadingOrder = xlContext  
.MergeCells = False  
  
End With  
  
Range("AB2").Select  
  
Selection.Copy  
  
Range("AB3:AB27354").Select  
  
ActiveSheet.Paste
```

End Sub

**A2.** Macro para calcular el índice de Pareto y parámetro de Hurst.

```
Sub Macro2()  
  
Sheets.Add
```

```

Sheets("Hoja1").Select
Sheets("Hoja1").Name = "HURST"
Sheets("resultadosi").Select
Columns("AE:AE").Select
Selection.Copy
Sheets("HURST").Select
Columns("A:A").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
End With
Range("A1").Select
ActiveCell.FormulaR1C1 = "BYTES "
Range("B1").Select

```

```

ActiveCell.FormulaR1C1 = "NEWBYTES"

Range("D1").Select

ActiveCell.FormulaR1C1 = "N"

Range("E1").Select

ActiveCell.FormulaR1C1 = "=COUNT(C[-4])"

Range("D2").Select

ActiveCell.FormulaR1C1 = "MEDIA BYTES"

Range("E2").Select

Columns("D:D").EntireColumn.AutoFit

ActiveCell.FormulaR1C1 = "=SUM(RC[-4]:R[27352]C[-4])/R[-1]C"

Columns("B:B").Select

With Selection

    .HorizontalAlignment = xlRight

    .VerticalAlignment = xlBottom

    .WrapText = False

    .Orientation = 0

    .AddIndent = False

    .IndentLevel = 0

    .ShrinkToFit = False

    .ReadingOrder = xlContext

    .MergeCells = False

End With

Range("B2").Select

ActiveCell.FormulaR1C1 = "0"

```

```

Range("B3").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+(R[-1]C[-1]-R2C5)"
Range("B3").Select
Selection.Copy
Range("B4:B27354").Select
ActiveSheet.Paste
Range("D3").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "DESVIACIÓN STD"
Range("E3").Select
Columns("D:D").EntireColumn.AutoFit
ActiveCell.FormulaR1C1 = "=STDEVP(R[-1]C[-4]:R[27351]C[-4])"
Range("D4").Select
ActiveCell.FormulaR1C1 = "MAXIMO NEW BYTES"
Range("D5").Select
Columns("D:D").EntireColumn.AutoFit
ActiveCell.FormulaR1C1 = "MINIMO NEW BYTES"
Range("E4").Select
ActiveCell.FormulaR1C1 = "=MAXIMO(R[-2]C[-3],R[-2]C[-3]:R[27350]C[-3],R[-
3]C)"
Range("E4").Select
Selection.Copy
Range("E5").Select
ActiveSheet.Paste

```

Application.CutCopyMode = False

ActiveCell.FormulaR1C1 = "=MINIMO(R[-3]C[-3],R[-3]C[-3]:R[27347]C[-3],R[-3]C)"

Range("E5").Select

ActiveCell.FormulaR1C1 = "=MINIMO(R[-3]C[-3],R[-3]C[-3]:R[27347]C[-3],R[-4]C)"

Range("E5").Select

ActiveCell.FormulaR1C1 = "=MINIMO(R[-3]C[-3],R[-3]C[-3]:R[27349]C[-3],R[-4]C)"

Range("D6").Select

ActiveCell.FormulaR1C1 = "RANGO"

Range("E6").Select

ActiveCell.FormulaR1C1 = "=R[-2]C-R[-1]C"

Range("D7").Select

ActiveCell.FormulaR1C1 = "R/S"

Range("E7").Select

ActiveCell.FormulaR1C1 = "=R[-1]C/R[-4]C"

Range("D8").Select

ActiveCell.FormulaR1C1 = "LOG10(R/S)"

Range("E8").Select

ActiveCell.FormulaR1C1 = "=LOG10(R[-1]C)"

Range("D9").Select

ActiveCell.FormulaR1C1 = "LOG10(R/S)"

Range("D9").Select

ActiveCell.FormulaR1C1 = "LOG10(N)"

Range("E9").Select

ActiveCell.FormulaR1C1 = "=LOG10(R[-8]C)"

```
Range("D10").Select
ActiveCell.FormulaR1C1 = "HURST"
Range("E10").Select
ActiveCell.FormulaR1C1 = "=R[-2]C/R[-1]C"
Range("E11").Select
End Sub
```

```
Function MAXIMO(A, Bi As Range, FIN)
```

```
For I = 2 To FIN
```

```
B = Bi(I, 1).Value
```

```
If B > A Then
```

```
A = B
```

```
End If
```

```
Next I
```

```
MAXIMO = A
```

```
End Function
```

```
Function MINIMO(A, Bi As Range, FIN)
```

```
For I = 2 To FIN
```

```
B = Bi(I, 1).Value
```

```
If B < A Then
```

```
A = B
```

```
End If
```

```
Next I
```

```
MINIMO = A
```



End Function

Function MEDIAINI(A, R As Range, f, n)

H = 0

suma = 0

m = f + 1

Var = f

For J = m To n

B = R(J, 1)

If A = B Then

C = R(Var, 2)

D = R(J, 2)

Var = J

dif = D - C

If dif <> 0 Then

H = H + 1

suma = suma + dif

'End If

End If

Next J

MEDIAINI = suma / H

End Function

Function MEDIAFILE(IP, Bi As Range, fil, num)

H = 1

```

Sumi = Bi(fil, 3)

mi = fil + 1

Var = fil

For Ji = mi To num

B = Bi(Ji, 1)

If IP = B Then

C = Bi(Var, 2)

D = Bi(Ji, 2)

Var = Ji

dif = D - C

If dif <> 0 Then

H = H + 1

E = Bi(Ji, 3)

Sumi = Sumi + E

End If

End If

Next Ji

MEDIAFILE = Sumi / H

End Function

```

**A3.** Macro para calcular la media del tamaño de archivo y la media del tiempo entre conexiones.

Sub Macro3()

ActiveWindow.SmallScroll ToRight:=10

Columns("V:V").Select

Selection.Copy

Sheets.Add

Sheets("Hoja1").Select

Sheets("Hoja1").Name = "MEDIA IS"

Sheets("MEDIA IS").Select

Columns("A:A").Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=False

Columns("A:A").EntireColumn.AutoFit

Application.CutCopyMode = False

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

```

Range("B1").Select
Sheets("resultadosi").Select
Columns("O:O").Select
Selection.Copy
Sheets("MEDIA IS").Select
Columns("B:B").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Columns("B:B").EntireColumn.AutoFit
Range("C1").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "=COUNT(C[-1])+1"
Columns("A:B").Select
Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Key2:=Range("B2") _
, Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:= _
False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2 _
:=xlSortNormal
Range("C2").Select
ActiveCell.FormulaR1C1 = "=ROW(RC[-2])"
Range("D2").Select
ActiveCell.FormulaR1C1 = _
"=IF(RC[-3]=R[-1]C[-3],0,MEDIAINI(RC[-3],R2C1:R27354C2,RC[-1],R1C3))"
Range("D1").Select
ActiveCell.FormulaR1C1 = "MEDIA"

```

Range("D1").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlBottom

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Range("C2:D2").Select

Selection.Copy

Range("C3:D27354").Select

Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, \_

SkipBlanks:=False, Transpose:=False

Sheets.Add

Sheets("Hoja2").Select

Sheets("Hoja2").Name = "SUB MEDIA IS"

Sheets("MEDIA IS").Select

Range("A:A,D:D").Select

Range("D1").Activate

Application.CutCopyMode = False

```

Selection.Copy

Sheets("SUB MEDIA IS").Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False

Application.CutCopyMode = False

Selection.Subtotal GroupBy:=1, Function:=xlSum, TotalList:=Array(2), _
Replace:=True, PageBreaks:=False, SummaryBelowData:=True

ActiveSheet.Outline.ShowLevels RowLevels:=2

Selection.SpecialCells(xlCellTypeVisible).Select

Selection.Copy

Sheets.Add

Sheets("Hoja3").Select

Sheets("Hoja3").Name = "TOTAL MEDIA IS"

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False

Columns("A:A").EntireColumn.AutoFit

Range("C1").Select

Application.CutCopyMode = False

ActiveCell.FormulaR1C1 = "=COUNT(C[-1])"

Range("C2").Select

Application.Goto Reference:="R103C3"

ActiveWindow.SmallScroll Down:=12

ActiveWindow.ScrollRow = 1

Range("D1").Select

```

End Sub

Sub Macro5()

Columns("V:V").Select

Selection.Copy

Sheets("resultadosi").Select

Sheets.Add

Sheets("Hoja1").Select

Sheets("Hoja1").Name = "MEDIA F"

Sheets("resultadosi").Select

Application.CutCopyMode = False

Selection.Copy

Sheets("MEDIA F").Select

Columns("A:A").Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=False

Sheets("resultadosi").Select

Columns("O:O").Select

Application.CutCopyMode = False

Selection.Copy

Sheets("MEDIA F").Select

Columns("B:B").Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

:=False, Transpose:=False

```

Sheets("resultadosi").Select

ActiveWindow.ScrollColumn = 12

ActiveWindow.ScrollColumn = 13

Columns("AE:AE").Select

Application.CutCopyMode = False

Selection.Copy

Sheets("MEDIA F").Select

Columns("C:C").Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False

Columns("A:C").Select

Application.CutCopyMode = False

With Selection

    .HorizontalAlignment = xlCenter

    .VerticalAlignment = xlBottom

    .WrapText = False

    .Orientation = 0

    .AddIndent = False

    .IndentLevel = 0

    .ShrinkToFit = False

    .ReadingOrder = xlContext

    .MergeCells = False

End With

Range("C1").Select

```



```

ActiveCell.FormulaR1C1 = "BYTES"

Range("D1").Select

ActiveCell.FormulaR1C1 = "=COUNT(C[-2])"

Range("D1").Select

ActiveCell.FormulaR1C1 = "=COUNT(C[-2])+1"

Columns("A:C").Select

Selection.Sort Key1:=Range("A2"), Order1:=xlAscending, Key2:=Range("B2") _
, Order2:=xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:= _
False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, DataOption2 _
:=xlSortNormal

Range("D2").Select

ActiveCell.FormulaR1C1 = "=ROW(RC[-3])"

Range("E1").Select

With Selection

    .HorizontalAlignment = xlCenter

    .VerticalAlignment = xlBottom

    .WrapText = False

    .Orientation = 0

    .AddIndent = False

    .ShrinkToFit = False

    .ReadingOrder = xlContext

    .MergeCells = False

End With

ActiveCell.FormulaR1C1 = "MEDIA F"

```

```

Range("E1").Select

ActiveCell.FormulaR1C1 = "MEDIA F BYTES"

Range("F1").Select

With Selection

    .HorizontalAlignment = xlCenter

    .VerticalAlignment = xlBottom

    .WrapText = False

    .Orientation = 0

    .AddIndent = False

    .ShrinkToFit = False

    .ReadingOrder = xlContext

    .MergeCells = False

End With

ActiveCell.FormulaR1C1 = "MEDIA F BITS"

Range("F2").Select

Columns("E:E").EntireColumn.AutoFit

Columns("F:F").EntireColumn.AutoFit

Range("E2").Select

ActiveCell.FormulaR1C1 = _

    "=IF(RC[-4]=R[-1]C[-4],0,MEDIAFILE(RC[-4],R1C1:R27354C2,RC[-1],R1C4))"

Range("F2").Select

ActiveCell.FormulaR1C1 = "=RC[-1]*8"

Range("D2:F2").Select

Selection.Copy

```

```
Range("D3:F27354").Select
```

```
ActiveSheet.Paste
```

```
End Sub
```