



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# Sistema de monitoreo de capacidad de redes inalámbricas 802.11

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE  
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Alejandro Álvarez, Martina Isbarbo, Bruno Rivas

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS  
PARA LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO ELECTRICISTA.

## TUTOR

Germán Capdehourat ..... Universidad de la República  
Federico La Rocca ..... Universidad de la República

## TRIBUNAL

Claudina Rattaro ..... Universidad de la República  
Gabriel Gómez ..... Universidad de la República  
Eduardo Grampín ..... Universidad de la República

Montevideo  
lunes 7 marzo, 2016

*Sistema de monitoreo de capacidad de redes inalámbricas 802.11*, Alejandro Álvarez, Martina Isbarbo, Bruno Rivas.

Esta tesis fue preparada en L<sup>A</sup>T<sub>E</sub>X usando la clase iietesis (v1.1).  
Contiene un total de 102 páginas.  
Compilada el lunes 7 marzo, 2016.  
<http://iie.fing.edu.uy/>

Somewhere, something incredible is waiting to be known.

CARL SAGAN

Esta página ha sido intencionalmente dejada en blanco.

# Agradecimientos

A nuestras familias y amigos que nos apoyaron en todo momento a lo largo de este año de trabajo...

A nuestros tutores que estuvieron siempre disponibles ante cualquier inquietud y cuyos consejos fueron fundamentales a la hora de la realización de este proyecto...

Al personal de Plan Ceibal que estuvo disponible para brindarnos su asistencia en los momentos en que la necesitamos...

A todo el personal de la Escuela N° 51 República del Paraguay por abrirnos las puertas de sus instalaciones y cuya amabilidad y disposición en cada una de nuestras visitas hizo que pudiéramos trabajar con total comodidad...

A todos ellos, ¡MUCHÍSIMAS GRACIAS!

Esta página ha sido intencionalmente dejada en blanco.

# Resumen

En este proyecto se busca obtener un indicador del desempeño de una red inalámbrica del Plan Ceibal y diseñar e implementar un sistema en software que permita su adquisición en tiempo real.

Se desarrolla un modelo matemático que a partir de las tasas de transmisión de capa física de los usuarios presentes en una red inalámbrica 802.11 estima cuál es el throughput TCP máximo de bajada que se puede alcanzar entre todos los usuarios. Este es luego implementado en software y validado en pruebas de laboratorio obteniendo errores relativos menores a 7%. A partir de este se proponen diversos modelos capaces de estimar cuál es el máximo throughput TCP que puede obtener un nuevo usuario que se conecte a la red a cursar tráfico de bajada. Estos son nuevamente implementados en software y evaluados en pruebas de laboratorio y pruebas de campo realizadas en una red de Plan Ceibal de una escuela pública. En esta nueva situación el modelo elegido obtuvo errores relativos menores a 10% en el 90% de los casos

Finalmente, una vez validados los modelos matemáticos, se desarrolla en base a estos una herramienta de monitoreo que permite visualizar en tiempo real la variación de los indicadores modelados.

Esta página ha sido intencionalmente dejada en blanco.



# Tabla de contenidos

<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>v</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del proyecto . . . . .	2
1.2. Objetivo y alcance del proyecto . . . . .	4
1.3. Estructura del documento . . . . .	5
<b>2. Marco teórico</b>	<b>7</b>
2.1. Protocolos y estándares . . . . .	7
2.1.1. TCP . . . . .	7
2.1.2. UDP . . . . .	9
2.1.3. SNMP . . . . .	10
2.1.4. PLCP . . . . .	11
2.1.5. IEEE 802.11 . . . . .	11
<b>3. Laboratorio</b>	<b>19</b>
3.1. Modelo de la red . . . . .	19
3.2. Configuración de la red . . . . .	21
3.3. Herramientas utilizadas . . . . .	23
3.3.1. Iperf . . . . .	23
3.3.2. Winbox . . . . .	23
3.3.3. Oracle VirtualBox VM . . . . .	23
3.3.4. WinSCP . . . . .	24
3.3.5. Manage Engine Free SNMP MIB Browser . . . . .	24
3.3.6. RRDtool . . . . .	24
<b>4. Primer modelo de capacidad</b>	<b>25</b>
4.1. Escenario e hipótesis . . . . .	25
4.2. Obtención del modelo . . . . .	26
4.3. Implementación del modelo . . . . .	29
4.4. Elección del tiempo de medición . . . . .	34
4.5. Pruebas y validación del modelo . . . . .	35

## Tabla de contenidos

<b>5. Segundo modelo de capacidad</b>	<b>41</b>
5.1. Escenario e hipótesis . . . . .	41
5.2. Obtención del modelo . . . . .	42
5.2.1. Escenario 1: No hay usuarios saturados. . . . .	43
5.2.2. Escenario 2: Hay $n$ usuarios saturados. . . . .	45
5.3. Implementación del modelo . . . . .	51
5.4. Pruebas en laboratorio y validación de modelos . . . . .	56
5.4.1. Escenario sin usuarios saturados . . . . .	58
5.4.2. Escenario en que hay $n$ usuarios saturados . . . . .	58
5.5. Pruebas en escuela . . . . .	62
5.6. Análisis de los resultados . . . . .	64
5.6.1. Escenario sin usuarios saturados . . . . .	64
5.6.2. Escenario en que hay $n$ usuarios saturados . . . . .	66
<b>6. Sistema de monitoreo</b>	<b>69</b>
6.1. Implementación . . . . .	69
<b>7. Conclusiones</b>	<b>73</b>
7.1. Conclusiones generales . . . . .	73
7.2. Trabajos futuros . . . . .	74
<b>A. Degradación del desempeño debido al multi-rate</b>	<b>75</b>
A.1. Análisis . . . . .	75
<b>B. Gestión de proyecto</b>	<b>79</b>
<b>C. Contenido del CD</b>	<b>81</b>
<b>Referencias</b>	<b>83</b>
<b>Índice de tablas</b>	<b>86</b>
<b>Índice de figuras</b>	<b>88</b>

# Capítulo 1

## Introducción

Una red de área local inalámbrica (o Wireless Local Area Network, más conocida por su sigla WLAN) es un sistema de comunicación inalámbrico que utiliza la radiofrecuencia para la transmisión de información presentándose así como alternativa a las redes cableadas. Durante los últimos años, y con el auge de los dispositivos móviles y laptops, las redes inalámbricas están tomando cada vez más importancia.

Los inicios de las redes de datos inalámbricas datan de 1979 cuando ingenieros de IBM en Suiza presentan los resultados de un experimento [1] que consistía en crear una red a través de enlaces infrarrojos. Los estudios progresaron tanto con tecnologías infrarrojas como con el uso de microondas hasta que en 1985 la FCC<sup>1</sup> (Federal Communications Commission) comenzó a regular el uso de las bandas de frecuencia ISM (Industrial, Scientific and Medical) propiciando así una mayor actividad en la industria y encaminando las redes inalámbricas al mercado.

En el año 1997, dada la necesidad de establecer un mecanismo de conexión que fuera compatible entre dispositivos de todos los fabricantes, la IEEE define la primera versión del estándar IEEE 802.11 [2] especificando las normas de funcionamiento de una WLAN infrarroja. En este también se define el protocolo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) que es de particular interés para el desarrollo de este proyecto. Con el transcurso del tiempo, y para ir adaptándose y evolucionando junto a las necesidades de los usuarios, el estándar pasó por varias revisiones con el objetivo de hacer un uso más eficiente del espectro radioeléctrico y así obtener mayores velocidades de transmisión. Entre las revisiones más destacadas se encuentran, entre otras, la 802.11a y 802.11b (ambas ratificadas en 1999). Estas utilizan el mismo juego de protocolos que el estándar original, pero operando en las bandas de 5 GHz y 2,4 GHz respectivamente y aumentando sus capacidades de transferencia. Dentro de las revisiones que

---

<sup>1</sup>La Federal Communications Commission, creada en 1943, es una agencia estatal de Estados Unidos encargada de la regulación de telecomunicaciones internacionales por radio, televisión, redes inalámbricas, teléfonos, satélites y cable.

## Capítulo 1. Introducción

ha tenido el estándar destacan también las revisiones 802.11g (2003) cuya novedad más importante respecto a versiones anteriores es un aumento de la velocidad de transferencia a 54Mbps, y 802.11n (2009) cuya principal novedad respecto a los estándares anteriores es que agrega el uso de antenas MIMO (Multiple Input - Multiple Output).

Las redes WLAN son de vital importancia para el cumplimiento de los objetivos del Plan Ceibal [3], por lo cual es fundamental garantizar su buen funcionamiento. Actualmente las redes del Plan Ceibal no disponen de una aplicación con la capacidad de evaluar el impacto que puede tener el ingreso de un nuevo usuario a la red y determinar así si se le niega o concede el acceso. Para poder tomar este tipo de decisiones es imprescindible conocer su comportamiento actual y en base a este poder estimar su comportamiento futuro.

### 1.1. Descripción del proyecto

A diferencia de lo que sucede en una red cableada, las redes inalámbricas una vez puestas en servicio suelen ver afectado su desempeño respecto al diseño original debido a factores tales como la cantidad de clientes conectados a la red, las características del enlace entre cada cliente y el AP (velocidad de transmisión, intensidad de la señal), interferencia producida por presencia de otras redes inalámbricas en el entorno u otras señales que se transmiten en la mismas bandas de frecuencia. El impacto de todos estos elementos varía fuertemente con el tiempo lo cual hace que el desempeño de la red sea muy difícil de predecir.

La motivación fundamental de este proyecto es monitorear datos en tiempo real que permitan conocer para cada instante de tiempo la calidad del servicio que se está brindando a los actuales y posibles futuros usuarios de la red. Para lograrlo, se desea implementar un software que, sin afectar el desempeño de la red, logre estimar un indicador del desempeño que a efectos del proyecto se define como *capacidad*.

Cuando se habla de *capacidad* se hace en términos de máximas velocidades de transferencia de datos en capa de transporte, o capa 4 del modelo TCP/IP. Según cual sea la necesidad y punto de vista del usuario o administrador de la red, resulta de interés diferenciar entre dos concepciones distintas de capacidad. La primera está asociada al throughput máximo al que se puede traficar simultáneamente en la red por todos los usuarios en su conjunto, mientras que la segunda está asociada al throughput que podría obtener un nuevo usuario que se conecte a la red.

Se plantean entonces dos definiciones distintas de capacidad para las cuales se elaboran modelos matemáticos, que se implementan en software y a partir de los cuales se estima la capacidad de la red:

## 1.1. Descripción del proyecto

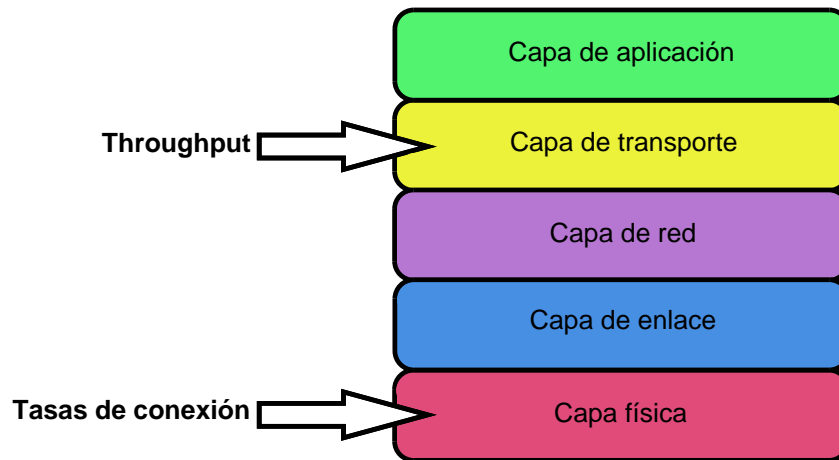


Figura 1.1: Modelo TCP/IP.

- **Definición 1:** Partiendo de un estado dado de la red en el cual se supondrá que solo se está cursando tráfico TCP, se definirá la capacidad de la red como el throughput TCP máximo que podrían traficar todos los usuarios en su conjunto, si cada uno de ellos se pusiera a descargar tanto como puede.
- **Definición 2:** Partiendo de un estado dado de la red en el que puede haber distintos tipos de tráfico en curso (mezcla de TCP y UDP tanto de bajada como de subida), se definirá la capacidad de la red como el throughput TCP máximo que podría obtener un nuevo usuario al ingresar a la red cursando únicamente tráfico de bajada.

A partir de ahora se referirá al modelo para estimar la capacidad según la primera definición como modelo 1, y al modelo para estimar la capacidad según la segunda definición como modelo 2. Cabe aclarar que estas dos situaciones a analizar fueron definidas por Plan Ceibal de acuerdo a su interés. La implementación y validación de dichos modelos en escenarios reales, con todas las complicaciones que esto conlleva, supone una gran dificultad.

Para realizar estas estimaciones los modelos contarán con información de las tasas de transmisión en capa física, los que se denominarán *tasas de conexión*, e información de los bytes cursados en capa de transporte por los distintos usuarios conectados al AP. A partir de estos datos se estimará el máximo throughput TCP según los modelos definidos. Haciendo un abuso del lenguaje se referirá al máximo throughput TCP simplemente como *máximo throughput*. En la figura 1.1 se presenta un esquema del modelo TCP/IP donde se indica a qué capa corresponden estos conceptos.

Existe vasta documentación acerca de modelos matemáticos que estimen la capacidad de la red para el modelo 1 (ver por ejemplo [4], [5], [6] y [7]). Sin embargo, son pocos los trabajos que estudian y proponen métodos para estimar el

## Capítulo 1. Introducción

throughput que podría obtener un posible nuevo usuario de la red. Existen diversas alternativas para abordar dicho problema tales como hacer medidas activas con algunos dispositivos desplegados en la red, o incluso tener otros equipos monitoreándola con medidas pasivas del aire. En [8] y [9] se proponen algunos métodos que estiman la capacidad disponible en base a información obtenida por los nodos de la red que es luego transmitida a nodos vecinos. Estos métodos son pasivos pues no introducen paquetes de prueba a la red. Sin embargo, su implementación es compleja y requiere que los nodos accedan al medio impactando de cierta forma en la capacidad de la red.

### 1.2. Objetivo y alcance del proyecto

El objetivo de este proyecto es obtener e implementar en software un modelo matemático que permita estimar la capacidad de una red inalámbrica a partir de cierta información obtenida de ella. Se trabajará únicamente con medidas pasivas obtenidas directamente de los propios APs ya instalados, lo cual abarata los costos y simplifica enormemente el despliegue ya que solo implica desarrollo de software que será ejecutado en los servidores que se encuentran directamente conectados a los APs.

En todos los casos, para la validación de los modelos, se realizarán pruebas en laboratorio con tráfico real y se contrastarán los resultados medidos con los estimados. Como objetivo específico se plantea obtener errores relativos menores al 10 %.

El modelo se diseñará para ser aplicado principalmente en redes del Plan Ceibal por lo cual se limitará a utilizar los equipos y parámetros de configuración allí utilizados. Sin embargo, realizando algunos ajustes, los resultados obtenidos pueden aplicarse a cualquier otra red inalámbrica que trabaje bajo el estándar 802.11.

El proyecto abarca el estudio de los protocolos y estándares utilizados, el análisis de los mecanismos definidos en estos, la obtención de los modelos matemáticos, el desarrollo de un software para estimar la capacidad de una red inalámbrica, la validación de los modelos propuestos en pruebas de laboratorio y en una red puntual del Plan Ceibal, y el análisis del desempeño del software desarrollado.

Se desea también implementar una herramienta de monitoreo que permita conocer minuto a minuto la capacidad de la red para los dos modelos descritos en la sección anterior.

## 1.3. Estructura del documento

En el capítulo 2 se describen los conceptos más importantes relacionados a los estándares y protocolos que resultan imprescindibles para lograr una total comprensión del trabajo realizado. Se hace especial hincapié en el estándar IEEE 802.11, sus diferentes versiones y su método de control de acceso al medio. También se describen los protocolos TCP, UDP, PLCP y SNMP.

En el capítulo 3 se describe la configuración de la red implementada para el laboratorio y se presentan todas las herramientas utilizadas para la implementación y ejecución de los scripts usados para validar los modelos de estimación de capacidad propuestos.

En el capítulo 4 se describe el escenario supuesto para la situación 1 descrita en la sección 1.1. Se explican las hipótesis de trabajo consideradas para este caso, a partir de las cuales se llega a un modelo matemático que será posteriormente validado en un entorno controlado de laboratorio. Se presentan también las configuraciones realizadas en los equipos, scripts implementados, pruebas realizadas, y análisis y validación de resultados referentes al modelo 1.

En el capítulo 5, luego de haber validado el primer modelo de capacidad, se proponen a partir de este varios modelos matemáticos capaces de estimar el throughput que puede obtener un nuevo usuario que se conecte a una red. Se describen aquí también las hipótesis de trabajo utilizadas para este escenario, las pruebas realizadas para su validación, y los resultados obtenidos junto con el análisis correspondiente. Se presentan también en este capítulo los resultados obtenidos tras la implementación del software en una red del Plan Ceibal.

En el capítulo 6 se describe la implementación de un sistema de monitoreo basado en los modelos de los capítulos anteriores y que puede ser utilizado por los administradores de la red para obtener información de esta en tiempo real.

Finalmente, en el capítulo 7 se presentan las conclusiones finales sobre el trabajo realizado y se plantean posibles trabajos futuros.

Esta página ha sido intencionalmente dejada en blanco.



# Capítulo 2

## Marco teórico

En este capítulo se describen los protocolos y estándares que hacen posible las comunicaciones a través de redes inalámbricas así como también un protocolo para la gestión de redes de datos que será utilizado a la hora de la implementación. Una vez comprendidas las características y funcionamiento de dichos protocolos será posible obtener los modelos que estimen las capacidades definidas en la sección 1.1.

### 2.1. Protocolos y estándares

Un estándar define un conjunto de reglas específicas que aseguran el correcto funcionamiento de una cierta tecnología, y es a su vez quien hace posible la amplia difusión de esta. Gracias a la existencia de los estándares hay interoperabilidad entre equipos de distintos fabricantes, lo que en definitiva es beneficioso para el consumidor final. La implementación de las redes locales inalámbricas no es ajena a ello. Esta es posible gracias a la existencia y el funcionamiento de diversos protocolos de distintos tipos. En esta sección se presentan los estándares y protocolos imprescindibles para la completa comprensión del presente estudio.

#### 2.1.1. TCP

En esta sección se describen algunas generalidades del protocolo TCP (Transmission Control Protocol) [10] haciendo especial hincapié en el sistema de reconocimiento de segmentos recibidos. Esto último es de gran interés ya que a la hora de hacer análisis de throughputs parte de la discusión se centra en la cantidad de nodos que compiten por el acceso al medio y, debido al funcionamiento de este protocolo, el nodo receptor también forma parte de esta disputa.

TCP es un protocolo de capa de transporte, o capa 4 del modelo OSI, capaz de garantizar la transmisión correcta y ordenada de información sobre una red IP o en su defecto el reporte de un error a la capa de aplicación del nodo origen. En la gran mayoría de las aplicaciones se utiliza TCP como protocolo de transporte

## Capítulo 2. Marco teórico

por lo que es de gran interés entender su funcionamiento. En particular, como se estableció en 1.1, el primer modelo de capacidad que se desarrollará utiliza como hipótesis que todo el tráfico que se cursa en la red es de tipo TCP.

Durante una transmisión en una red IP los paquetes pueden perderse, duplicarse o llegar en distinto orden al que fueron enviados. TCP detecta estos problemas, solicita la retransmisión de los datos perdidos, reordena la información recibida y ayuda a minimizar la congestión de la red para disminuir este tipo de problemas.

El protocolo también toma la información a transmitir y la divide en partes, agregándole a cada una de ellas un encabezado formando segmentos TCP, que son luego encapsulados en datagramas a nivel IP. El encabezado de TCP, sin contar campos opcionales, tiene un largo de 20 bytes. En la figura 2.1 se presentan los campos que lo componen.

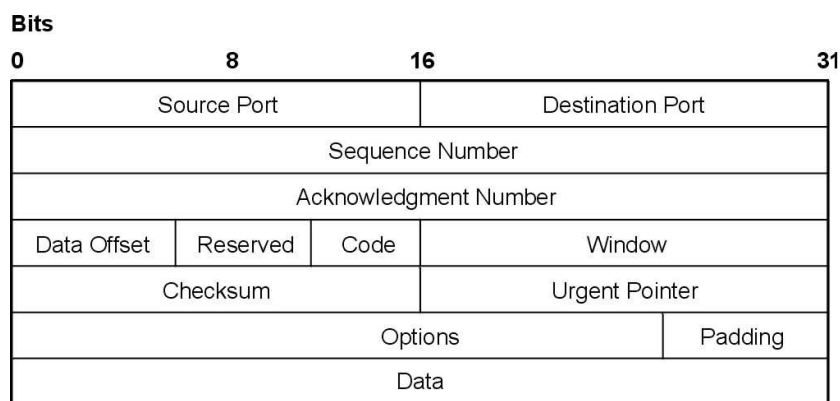


Figura 2.1: Paquete TCP.

Las transmisiones TCP se dividen en tres fases. La primera de ellas es el establecimiento de la conexión, luego la transmisión de los datos y por último la liberación de la conexión. Para el establecimiento de la conexión se utiliza el procedimiento llamado “negociación en tres pasos” (three-way handshake), mientras que para la desconexión se implementa un corte independiente para cada una de las vías.

TCP utiliza un número de secuencia para identificar cada byte de información. Dicho número identifica el orden en el que son transmitidos los bytes de forma que se pueda reconstruir la información correctamente, más allá de que se hayan fragmentado, desordenado o perdido segmentos en la red. En las primeras dos instancias de la “negociación en tres pasos” se intercambia el número de secuencia inicial.

El receptor debe enviarle una confirmación de los paquetes recibidos al transmisor. Para ello TCP implementa un esquema de reconocimientos en el que el

## 2.1. Protocolos y estándares

receptor envía un segmento “ACK” con el número de secuencia correspondiente al próximo byte que espera recibir, reconociendo de esta forma la correcta recepción de todos los bytes previos. Contar únicamente con este esquema de reconocimiento es ineficiente, ya que no se puede reconocer la llegada de un byte con número de secuencia posterior a otro perdido. Es por esto que en la RFC 2018 [11] se define la utilización de “SACK” (Selective Acknowledgment) lo que permite reconocer un rango específico de números de secuencia.

TCP utiliza un protocolo de control de flujo de ventana deslizante para evitar que el transmisor envíe información demasiado rápido como para que el receptor la pueda recibir y procesar correctamente. En cada segmento TCP el receptor especifica, en el campo *Window* del encabezado, la cantidad de información adicional que puede recibir (en bytes). El transmisor no puede enviar más que esa cantidad de información hasta no haber recibido un reconocimiento y una actualización del valor de la ventana.

### 2.1.2. UDP

UDP (User Datagram Protocol) es un protocolo de capa de transporte basado en el intercambio de datagramas. A diferencia de TCP, UDP no requiere un establecimiento de conexión previo al envío de datagramas. A su vez, las comunicaciones con este protocolo no utilizan confirmación de entrega o recepción, ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros. Como no ofrece garantías en cuanto a la correcta entrega de un mensaje se dice que es un protocolo “best effort”. Se usa principalmente en transmisiones de audio o video en tiempo real donde las retransmisiones no son una opción viable debido a su poca tolerancia a retardos. Se utiliza además para protocolos DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name System) y otros, en donde los intercambios de paquetes de conexión y desconexión son mayores que la información útil a transmitir.

El encabezado UDP consta de 8 bytes, compuesto por cuatro campos de 2 bytes cada uno, donde se indican los puertos de origen y destino, la longitud del mensaje a transmitir y una suma de verificación. Esto se muestra en la figura 2.2.

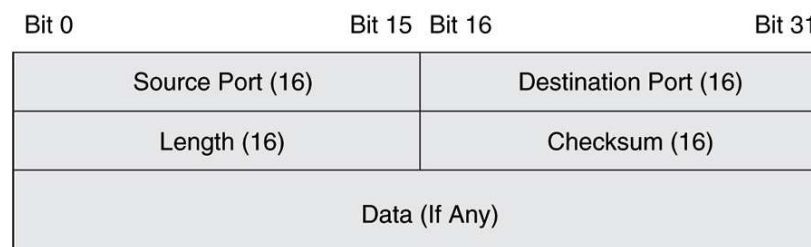


Figura 2.2: Datagrama UDP.

### 2.1.3. SNMP

El protocolo SNMP (Simple Network Management Protocol) es de gran importancia para la realización de este proyecto. Mediante su uso se obtendrá toda la información relacionada al tráfico cursado por los nodos en la red. Se trata de un protocolo de capa de aplicación (capa 7 del modelo OSI) que posibilita el intercambio de información de gestión de equipos de red. Estos equipos pueden ir desde routers y switches hasta una gran variedad de dispositivos que soporten IP.

A través de este protocolo se pueden hacer consultas a los equipos por gran variedad de métricas como ser: estado de utilización de CPU y memoria, temperatura, tráfico a través de sus interfaces, entre muchas otras. Esto hace que casi todos los sistemas de gestión de redes utilicen el protocolo SNMP para monitorear el estado de la red.

Para realizar las consultas de administración, SNMP utiliza un protocolo no orientado a conexión (UDP) para enviar grupos de mensajes, denominadas PDUs (Protocol Data Units), entre el sistema gestor y los equipos gestionados. Esto garantiza que las tareas de gestión no impactarán en el rendimiento global de la red pues se evita el uso de mecanismos de control, recuperación, o retransmisión de paquetes.

En la actualidad existen 3 versiones del protocolo, que no poseen compatibilidad entre ellas. Estas son:

- **SNMP v1:** Implementación original del protocolo.
- **SNMP v2:** Implementa mejoras en la performance y la seguridad e introduce nuevas funciones como *GetBulkRequest*, para obtener grandes cantidades de información en una sola consulta. La variante SNMP v2c (definida en la RFC 1901 [12]) utiliza como método de seguridad un esquema basado en comunidades que en definitiva actúan como claves para la autenticación. Es decir, si en la consulta SNMP al equipo se incluye la comunidad con la cual fue configurado, este responderá correctamente. De lo contrario no se obtendrá respuesta alguna.
- **SNMP v3:** Implementa mejoras adicionales en cuanto a la seguridad. Sin embargo, su difusión aún no es demasiado amplia.

En todas las implementaciones de este proyecto se utiliza la versión 2c de SNMP debido a que es la más difundida en la industria y permite de manera muy sencilla agregarle cierta seguridad a la implementación.

El protocolo no define qué información puede ser consultada a cada equipo. Esto lo determina cada fabricante definiendo la MIB (Management Information Base) de sus equipos en donde asigna de forma jerárquica, en una estructura tipo árbol como la que se muestra en la figura 2.3, un OID (Object Identifier) a cada

## 2.1. Protocolos y estándares

variable que puede ser consultada. Existen MIBs estándar, comunes a todos los fabricantes, entre las cuales destacan por su gran utilidad<sup>1</sup>, la MIB-II (definida en la RFC 1213 [13]) y la SNMPv2-MIB (definida en la RFC 3418 [14]).

### 2.1.4. PLCP

El protocolo PLCP (Physical Layer Convergence Protocol) define una subcapa de la capa física del modelo OSI que actúa como interfaz entre la capa MAC (Medium Access Control) y la subcapa PMD (Physical Medium Dependent). Su objetivo es que las diferentes especificaciones de la capa física, tales como las tasas de transmisión, sean transparentes para la capa MAC, permitiendo de esta manera la sincronización entre transmisor y receptor. Para ello se añaden a las MPDUs a transmitir (MAC Protocol Data Units, o tramas MAC) un cierto preámbulo ( $PLCP_{preamble}$ ) y encabezado ( $PLCP_{header}$ ), como se muestra en la figura 2.4, que serán transmitidos a una tasa baja predefinida (basic rate). El estándar 802.11 se refiere a esta nueva trama como PPDU (PLCP Protocol Data Unit).

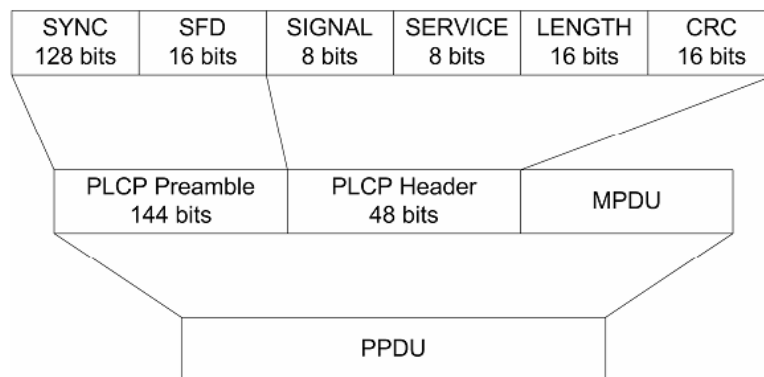


Figura 2.4: Composición de la trama PPDU. Imagen extraída de [15].

### 2.1.5. IEEE 802.11

En esta sección se describe el estándar IEEE 802.11, cuyo objetivo es establecer una serie de especificaciones sobre la capa física y el mecanismo de control de acceso al medio para implementar una red inalámbrica local. Se hará especial hincapié en el control de acceso al medio, puesto que su comprensión es de vital importancia para entender los análisis aquí presentados. Se introducirán también las revisiones del estándar que se consideran de interés para este estudio comentando brevemente sus principales similitudes y diferencias.

<sup>1</sup>Contienen los OIDs de las métricas monitoreadas con más frecuencia.

## Capítulo 2. Marco teórico

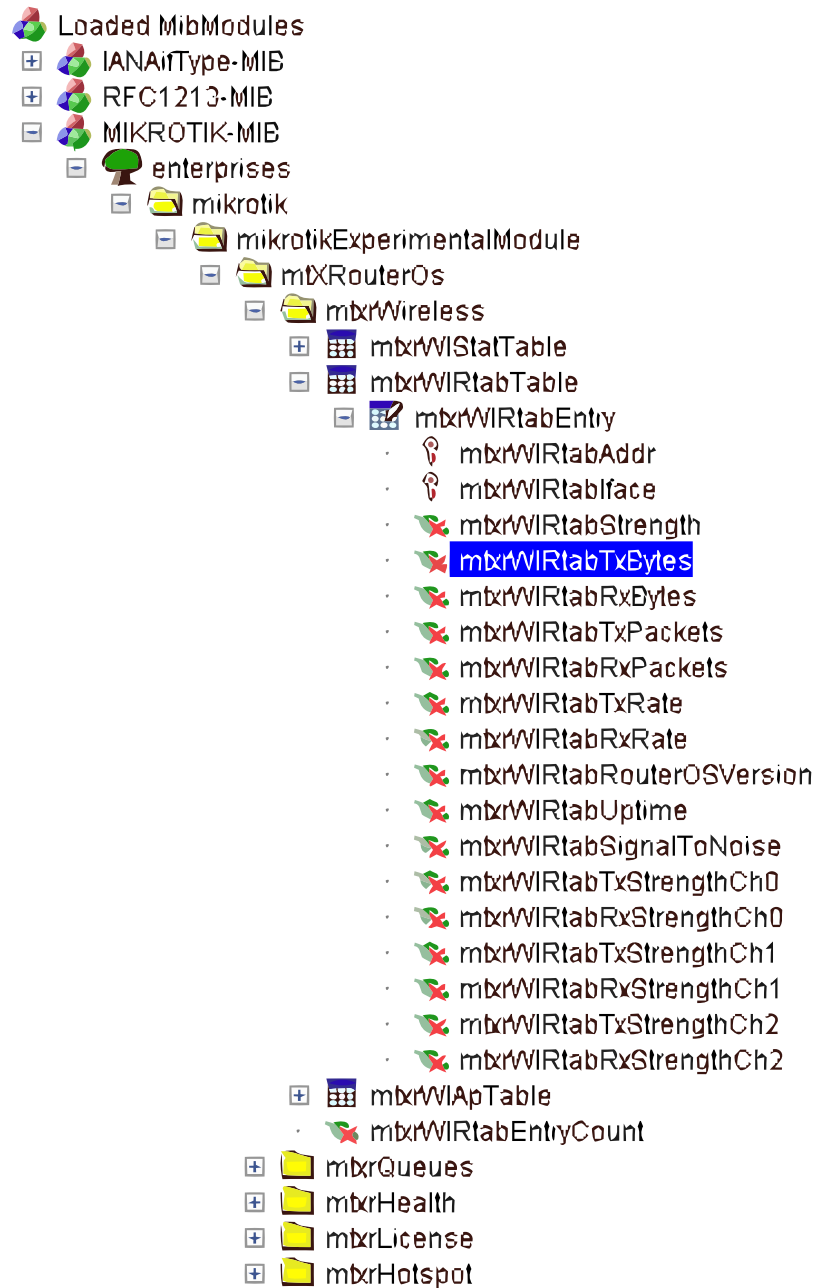


Figura 2.3: Ejemplo de una MIB cargada con un explorador de MIBs en donde puede verse la estructura tipo árbol. El OID seleccionado corresponde al de los bytes transmitidos hacia cada nodo conectado.

## 2.1. Protocolos y estándares

IEEE 802.11 es una serie de especificaciones de la capa física y de control de acceso al medio (MAC) creadas y mantenidas por la IEEE para implementar comunicaciones inalámbricas en las bandas de frecuencia de 2.4, 3.6, 5 y 60 GHz. A lo largo de los años se han ido desarrollando mejoras sobre el protocolo base por lo cual actualmente se dispone de una gran cantidad de revisiones (siendo las más conocidas las revisiones a, b, g, n, y ac). Algunas de estas son simplemente correcciones de revisiones anteriores, mientras que otras implementan determinadas características funcionales que posibilitan mayores velocidades de transmisión haciendo que destaquen frente al resto.

El estándar define para la capa física tanto las bandas de frecuencias como los distintos tipos de modulación empleados en cada versión y es allí, como se verá luego, donde radican las principales diferencias entre las distintas revisiones.

### Acceso al medio

Para el control de acceso al medio el estándar define el mecanismo DCF (Distributed Coordination Function). Este es un mecanismo de CSMA/CA ejecutado por cada uno de los nodos pertenecientes a la red al intentar acceder al medio. Su objetivo es evitar que hayan 2 nodos accediendo al canal de manera simultánea y de esta manera evitar que haya colisiones de paquetes en la red.

DCF es un mecanismo de 2 pasos. El primero de ellos, como todo protocolo CSMA, consiste en sensar el canal para detectar la presencia o no de alguna portadora. Esto se implementa a nivel de capa física, determinando si un cierto nivel de energía es sobrepasado por alguna señal de las mismas características que la que se desea transmitir.

Por otro lado, dado que en un medio inalámbrico es imposible detectar las colisiones de paquetes, lo que se busca es evitarlas, siendo este el segundo paso del mecanismo (CA). Esto se hace básicamente retrasando la transmisión de paquetes un tiempo aleatorio luego de haber sensado libre el canal. De esta manera se logra disminuir considerablemente la probabilidad de transmisiones simultáneas además de brindar el acceso al medio de manera equiprobable entre los distintos nodos.

El control de acceso al medio en su totalidad funciona de la siguiente manera. Luego de haber transmitido correctamente un paquete, y haber recibido el reconocimiento correspondiente, el nodo sensa el canal durante un tiempo DIFS (DCF Interframe Space). Si durante ese tiempo el canal está ocupado, se posterga la transmisión. Si de lo contrario el canal estuvo libre, se elige un valor aleatorio  $W_0$ , entre 0 y  $W - 1$ . Una vez elegido dicho valor, este decremента en uno por cada período de tiempo aTimeSlot en el que el medio se siga encontrando libre.

Si en algún momento el canal llega a encontrarse ocupado el contador se detiene hasta volver a detectar el canal libre durante un tiempo DIFS, momento en el cual

## Capítulo 2. Marco teórico

vuelve a decrementar en uno. Una vez que el contador llega a cero, el nodo verifica la existencia de algún paquete en la cola de transmisión. De haberlo, lo transmite inmediatamente. En caso contrario, espera hasta la llegada de un nuevo paquete para transmitir. Una vez llegado el paquete verifica que el medio haya estado libre por un tiempo mayor a DIFS, en cuyo caso realiza la transmisión inmediatamente. De lo contrario espera hasta que el canal esté libre por un tiempo mayor a DIFS y elige un nuevo  $W_0$  comenzando nuevamente el proceso de conteo.

Tras recibir correctamente el paquete, el nodo destino debe enviar un reconocimiento al nodo origen. Este reconocimiento, denominado 802.11 ACK, es transmitido a una tasa baja predefinida (basic rate) un tiempo SIFS (Short Interframe Space) luego de la recepción. Cabe destacar que dicho tiempo SIFS es menor que DIFS para darle mayor prioridad al reconocimiento. De esta manera el reconocimiento será siempre enviado inmediatamente después del paquete original. Si transcurre un tiempo ACK Timeout desde la transmisión sin que el nodo origen reciba dicho reconocimiento, intentará retransmitir el paquete.

Para la primera transmisión, y luego de cada transmisión satisfactoria, el valor  $W$  toma el valor  $W_{min}$ . Por cada retransmisión dicho valor  $W$  se duplica hasta llegar al límite  $W_{max}$ , momento a partir del cual  $W$  deja de crecer. La trama se descarta luego de cierta cantidad de retransmisiones.

En la figura 2.5 se muestra el diagrama de flujo del mecanismo recién descrito y en la figura 2.6 se muestra un ejemplo de diagrama temporal donde tres nodos intentan acceder al canal con un paquete listo para transmitir.

### Revisiones del estándar

A los efectos de este proyecto resultan interesantes las revisiones b y g del estándar debido a que son las más utilizadas en las redes inalámbricas desplegadas por Plan Ceibal.

- **802.11b:** fue desarrollada en 1999 y ofrece una velocidad máxima de transmisión de 11 Mbps. Utiliza CSMA/CA y modulación DSSS (Direct-sequence Spread Spectrum). Las tasas de conexión soportadas por esta versión son 1 Mbps, 2 Mbps, 5.5 Mbps y 11 Mbps.
- **802.11g:** fue desarrollada en 2003 y ofrece una velocidad máxima de transmisión de 54 Mbps. Utiliza modulación DSSS u OFDM (Orthogonal Frequency-division Multiplexing). Es compatible con 802.11b y utiliza las mismas bandas de frecuencia. Sin embargo, dado que 802.11b no soporta modulación OFDM, la presencia de equipos operando bajo el estándar b en redes que soportan 802.11g degradará considerablemente la máxima velocidad de transmisión. Las tasas de conexión soportadas por esta versión son 6 Mbps, 9 Mbps, 12 Mbps, 18 Mbps, 24 Mbps, 36 Mbps, 48 Mbps y 54 Mbps



## 2.1. Protocolos y estándares

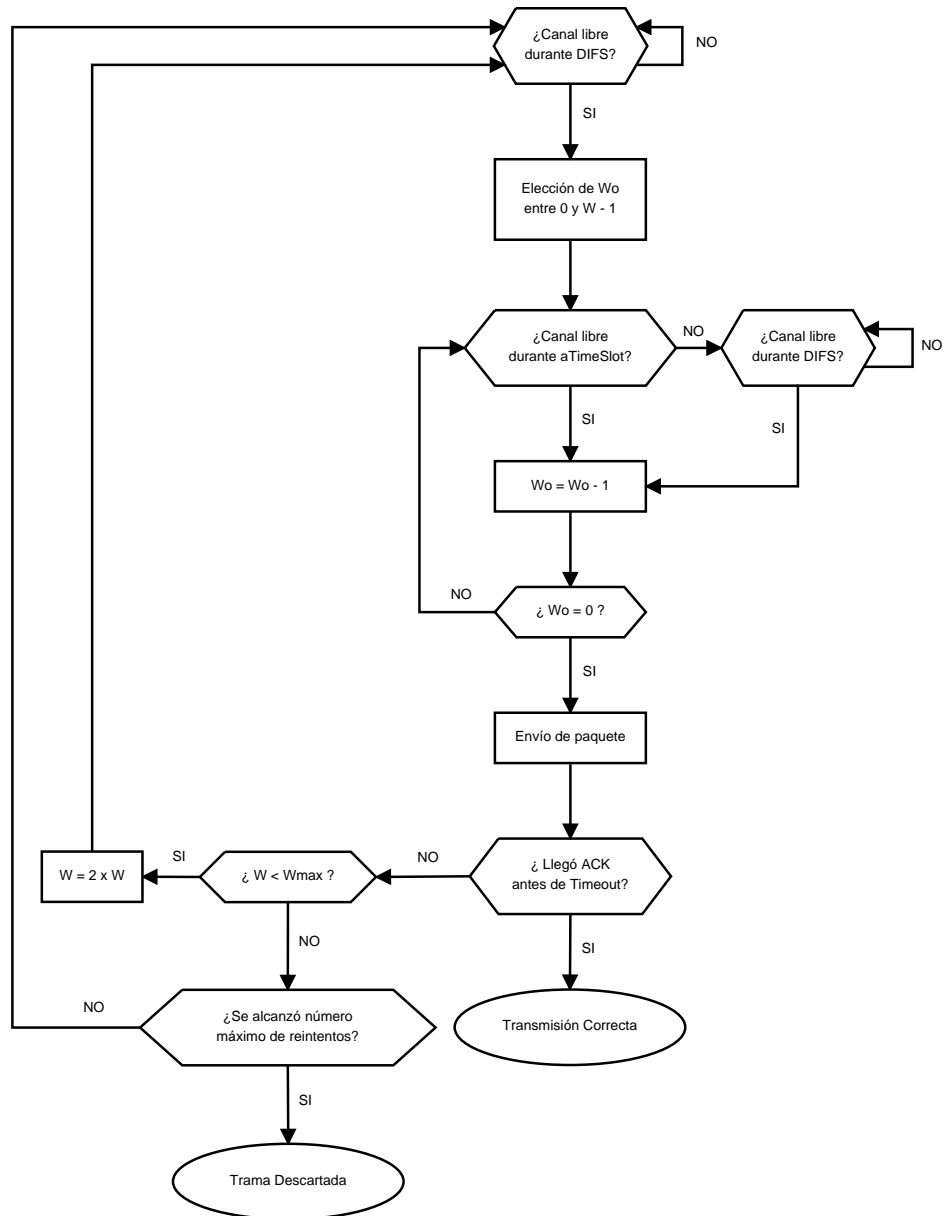


Figura 2.5: Diagrama de flujo de CSMA/CA.

## Capítulo 2. Marco teórico

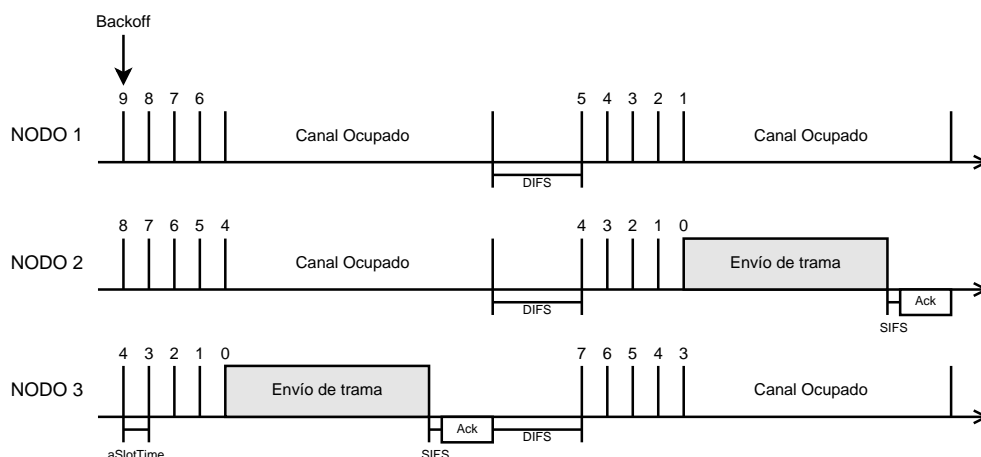


Figura 2.6: Ejemplo de diagrama temporal del mecanismo CSMA/CA en el que tres nodos siempre tienen un paquete listo para transmitir.

En la tabla 2.1 se muestran los parámetros definidos por el estándar para dichas revisiones.

Parámetro	802.11b	802.11g
aSlotTime ( $\mu s$ )	20	9 o 20
SIFS ( $\mu s$ )	10	10
DIFS ( $\mu s$ )	50	28 o 50
$W_{min}$	31	15
$W_{max}$	1023	1023

Tabla 2.1: Parámetros de los estándares 802.11b y 802.11g.

La elección del tiempo aSlotTime al trabajar con 802.11g depende de si el preámbulo PLCP agregado la trama es corto o largo. La diferencia entre ambos está en el tamaño del campo de sincronización dentro de dicho preámbulo. Los preámbulos cortos solo pueden ser empleados en redes que contengan exclusivamente equipos 802.11g, y su gran beneficio es una mejor eficiencia en el funcionamiento del estándar, principalmente en aplicaciones en tiempo real como ser streaming de video o VoIP.

### Automatic Rate Fallback

En una red inalámbrica las condiciones del medio pueden variar sensiblemente debido a factores tales como interferencias de otras redes inalámbricas, señales de otros dispositivos en la misma banda de frecuencia o por el simple movimiento de los clientes. Estos efectos pueden provocar que la SNR (Relación Señal a Ruido)

## 2.1. Protocolos y estándares

de la señal recibida sea baja, haciendo necesario trabajar con tasas de transmisión y recepción menores. De esta manera se hacen más robustas las comunicaciones frente a condiciones adversas y se evitan posibles errores en la transmisión de los paquetes.

Automatic Rate Fallback (ARF) [16] es el primer algoritmo publicado de adaptación de tasas de conexión. Su función es la de optimizar el throughput posible entre dos dispositivos mediante la adaptación de las tasas de recepción y transmisión de datos según las condiciones del medio. El algoritmo es sencillo y fácil de implementar, razón por la cual ha sido muy adoptado por los fabricantes de equipos que funcionan bajo el estándar 802.11.

El funcionamiento completo es el siguiente: cuando algunas transmisiones consecutivas (por ejemplo 2) fallan, es decir que el nodo que envía el paquete no recibe el ACK correspondiente, ARF disminuye la tasa de transmisión a la inmediatamente inferior (si es que lo hay) e inicia una cuenta regresiva. Si dicha cuenta llega a cero, o se efectúan un cierto número (por ejemplo 10) de transmisiones correctas, el nodo envía el siguiente paquete a la tasa inmediatamente superior y el contador es reiniciado. Si se recibe correctamente el ACK para este último paquete, el nodo se mantiene con la nueva tasa, de lo contrario vuelve a su tasa original en lugar de reintentar la transmisión.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 3

## Laboratorio

Para la validación, ajustes y eventuales correcciones de los modelos de capacidad que se propondrán en los siguientes capítulos es necesario contar con un laboratorio donde los modelos puedan evaluarse en un entorno controlado bajo distintos escenarios.

En este capítulo se describe la configuración de la red implementada en el laboratorio presentando a su vez las herramientas utilizadas para tal efecto, así como también aquellas utilizadas para la obtención de medidas y su posterior procesamiento. Como se mencionó anteriormente, la principal motivación de este proyecto es la optimización de las redes del Plan Ceibal, por lo cual las configuraciones y equipos utilizados en la red de laboratorio intentan ser las mismas que las allí utilizadas.

### 3.1. Modelo de la red

Para las pruebas de laboratorio se configuró una red básica, cuya topología consiste en diversos nodos que se conectan a un AP detrás del cual se encuentra un servidor local. Esto se muestra en el esquema de la figura 3.1.

Para que los resultados obtenidos en el laboratorio se asemejen a lo que eventualmente se obtendría en una red del Plan Ceibal, se utilizaron como nodos a conectar al AP algunos de los modelos de laptops distribuidos por Plan Ceibal a los estudiantes y profesores, y utilizados en los distintos centros educativos. Se utilizaron los modelos XO 4.0, Magallanes MG1, Magallanes MG2 y Magallanes MG4 que se muestran en la figura 3.2 y que fueron prestados por Plan Ceibal. El modelo XO 4.0 corre un sistema operativo basado en Fedora mientras que los tres modelos de Magallanes corren una distribución basada en Ubuntu. Las especificaciones técnicas completas de cada uno de estos equipos pueden ser consultadas en [17].

A continuación, en la figura 3.3, se muestra una foto de la red implementada

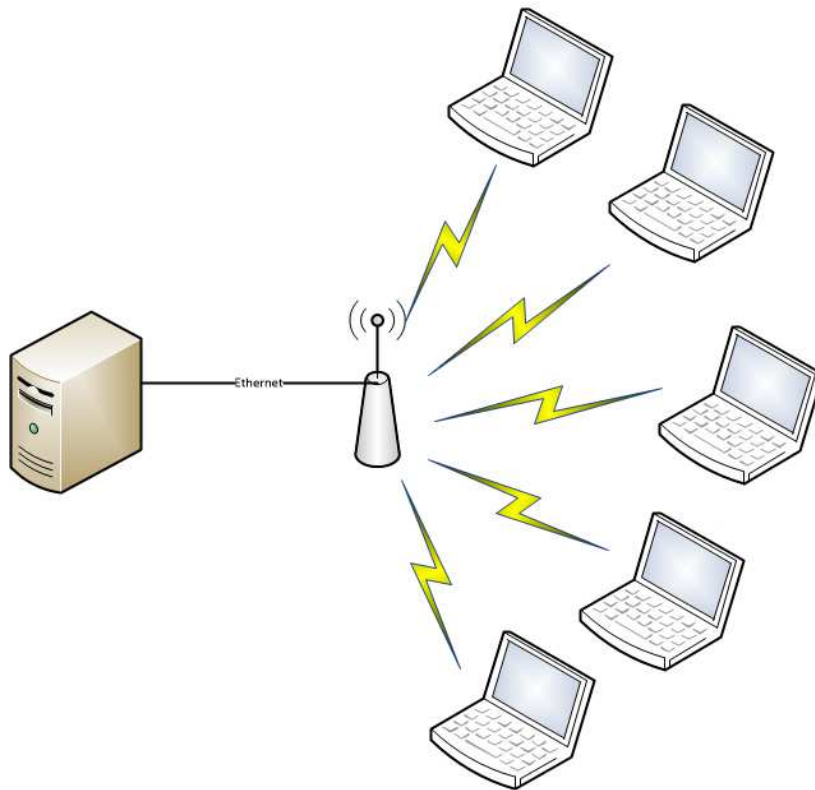


Figura 3.1: Diagrama de la red implementada.



Figura 3.2: Modelos de laptops del Plan Ceibal utilizados.

## 3.2. Configuración de la red



Figura 3.3: Red de laboratorio

para las pruebas en el laboratorio donde se puede ver el servidor (tercera laptop empezando de abajo a la izquierda) y la distribución real de los nodos alrededor del Access Point. Con esta disposición se intenta que todos los equipos tengan la misma potencia de señal recibida y por consiguiente las mismas tasas de conexión.

## 3.2. Configuración de la red

Se utilizó un Access Point MikroTik con placa base RouterBoard 433AH [18]. Dicha placa dispone de tres puertos Ethernet y tres puertos mini-PCI a los cuales se conectan dos placas RouterBoard R52n [19]. Estas proveen conectividad inalámbrica bajo los estándares 802.11 a/b/g/n tanto en banda de 2.4 GHz como de 5 GHz, soportando velocidades físicas de hasta 300 Mbps tanto en subida como en bajada. Cabe destacar que este AP fue provisto por el Instituto de Ingeniería Eléctrica y es el mismo que se utiliza en algunas redes de Plan Ceibal.

Para la configuración del AP se utilizó la aplicación WinBox, desarrollada por MikroTik para la gestión de sus equipos que corren bajo el sistema operativo RouterOS. A pesar de que los modelos serán implementados para trabajar tanto con 802.11b como con 802.11g, el AP fue configurado en modo *only-g* debido a que con esta configuración se prevé que se obtengan mayores variaciones de tasas, haciendo el análisis más enriquecedor. A su vez, las tasas habilitadas en la red configurada fueron las siguientes:

- **Tasas soportadas:** 6 Mbps, 9 Mbps, 12 Mbps, 18 Mbps, 24 Mbps, 36 Mbps, 48 Mbps, 54 Mbps

## Capítulo 3. Laboratorio

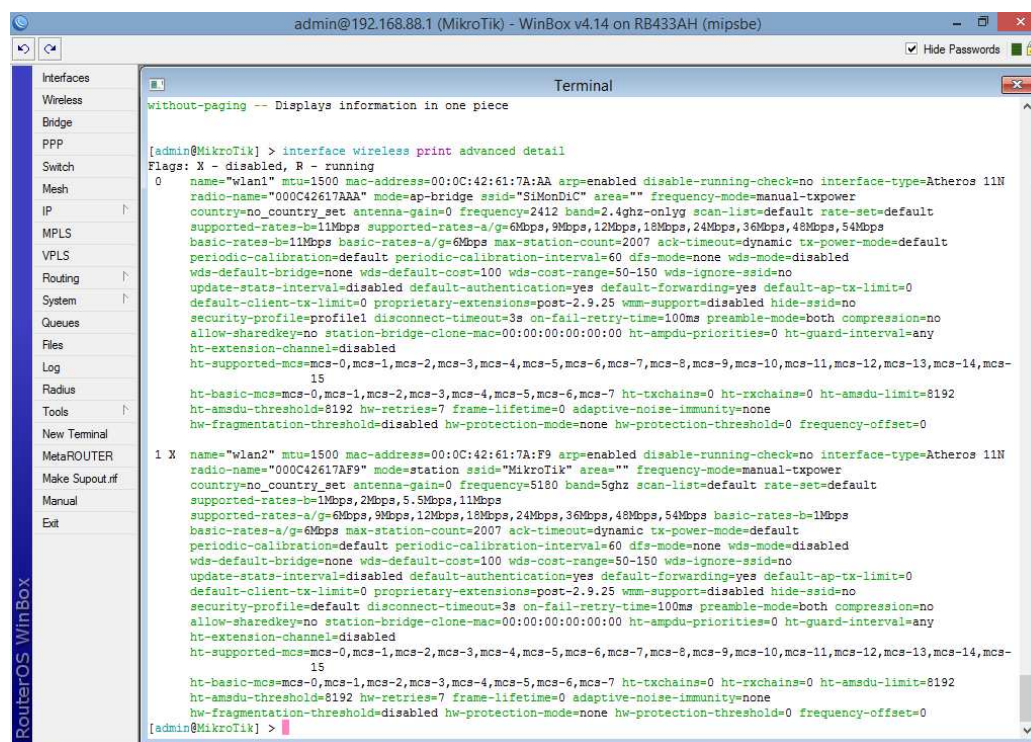
- **Basic Rate:** 6 Mbps

Es importante mencionar que las redes de Plan Ceibal utilizan las siguientes tasas:

- **Tasas soportadas:** 24 Mbps, 36 Mbps, 48 Mbps, 54 Mbps
- **Basic Rate:** 24 Mbps

Esto se hace de tal forma de limitar el acceso de usuarios lentos, evitando de esta manera degradar el rendimiento de la red debido a la presencia de nodos compitiendo por el canal y conectados a tasas bajas. Este efecto se explica en el apéndice A. Sin embargo, esta diferencia en la configuración de las tasas no trae problemas a la hora de las pruebas o de la validación del modelo puesto que el script implementado consulta periódicamente y se adapta a las variaciones de las tasas de conexión como se explica más adelante en la sección 4.3.

En la figura 3.4 se puede ver la configuración completa de las interfaces inalámbricas en Winbox.



```
admin@192.168.88.1 (MikroTik) - WinBox v4.14 on RB433AH (mipsbe)
Interfaces
Wireless
Bridge
PPP
Switch
Mesh
IP
MPLS
VPLS
Routing
System
Queues
Files
Log
Radius
Tools
New Terminal
MetaROUTER
Make Support.rtf
Manual
Exit

Terminal
without-paging -- Displays information in one piece

[admin@MikroTik] > interface wireless print advanced detail
Flags: X - disabled, R - running
0
name="wlan1" mtu=1500 mac-address=00:0C:42:61:7A:AA arp=enabled disable-running-check=no interface-type=Atheros 11N
radio-name="000C42617AAA" mode=ap-bridge ssid="SiMonDiC" area="" frequency-mode>manual-txpower
country=no_country_set antenna-gain=0 frequency=2412 band=2.4ghz-only scan-list=default rate-set=default
supported-rates-b=11Mbps supported-rates-a/g=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps
basic-rates-b=11Mbps basic-rates-a/g=6Mbps max-station-count=2007 ack-timeout=dynamic tx-power-mode=default
periodic-calibration=default periodic-calibration-interval=60 dfs-mode=none wds-mode=disabled
wds-default-bridge=none wds-default-cost=100 wds-cost-range=50-150 wds-ignore-ssid=no
update-stats-interval=disabled default-authentication=yes default-forwarding=yes default-ap-tx-limit=0
default-client-tx-limit=0 proprietary-extensions=post-2.9.25 wmm-support=disabled hide-ssid=no
security-profile=profile1 disconnect-timeout=3s on-fail-retry-time=100ms preamble-mode=both compression=no
allow-sharedkey=no station-bridge-clone-mac=00:00:00:00:00:00 ht-ampdu-priorities=0 ht-guard-interval=any
ht-extension-channel=disabled
ht-supported-mcs=mcs-0,mcs-1,mcs-2,mcs-3,mcs-4,mcs-5,mcs-6,mcs-7,mcs-8,mcs-9,mcs-10,mcs-11,mcs-12,mcs-13,mcs-14,mcs-15
ht-basic-mcs=mcs-0,mcs-1,mcs-2,mcs-3,mcs-4,mcs-5,mcs-6,mcs-7 ht-txchains=0 ht-rxchains=0 ht-amsdu-limit=8192
ht-amsdu-threshold=8192 hw-retries=7 frame-lifetime=0 adaptive-noise-immunity=none
hw-fragmentation-threshold=disabled hw-protection-mode=none hw-protection-threshold=0 frequency-offset=0

1 X
name="wlan2" mtu=1500 mac-address=00:0C:42:61:7A:F9 arp=enabled disable-running-check=no interface-type=Atheros 11N
radio-name="000C42617AF9" mode=station ssid="MikroTik" area="" frequency-mode>manual-txpower
country=no_country_set antenna-gain=0 frequency=5180 band=5ghz scan-list=default rate-set=default
supported-rates-b=1Mbps,2Mbps,5.5Mbps,11Mbps
supported-rates-a/g=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,54Mbps basic-rates-b=1Mbps
basic-rates-a/g=6Mbps max-station-count=2007 ack-timeout=dynamic tx-power-mode=default
periodic-calibration=default periodic-calibration-interval=60 dfs-mode=none wds-mode=disabled
wds-default-bridge=none wds-default-cost=100 wds-cost-range=50-150 wds-ignore-ssid=no
update-stats-interval=disabled default-authentication=yes default-forwarding=yes default-ap-tx-limit=0
default-client-tx-limit=0 proprietary-extensions=post-2.9.25 wmm-support=disabled hide-ssid=no
security-profile=default disconnect-timeout=3s on-fail-retry-time=100ms preamble-mode=both compression=no
allow-sharedkey=no station-bridge-clone-mac=00:00:00:00:00:00 ht-ampdu-priorities=0 ht-guard-interval=any
ht-extension-channel=disabled
ht-supported-mcs=mcs-0,mcs-1,mcs-2,mcs-3,mcs-4,mcs-5,mcs-6,mcs-7,mcs-8,mcs-9,mcs-10,mcs-11,mcs-12,mcs-13,mcs-14,mcs-15
ht-basic-mcs=mcs-0,mcs-1,mcs-2,mcs-3,mcs-4,mcs-5,mcs-6,mcs-7 ht-txchains=0 ht-rxchains=0 ht-amsdu-limit=8192
ht-amsdu-threshold=8192 hw-retries=7 frame-lifetime=0 adaptive-noise-immunity=none
hw-fragmentation-threshold=disabled hw-protection-mode=none hw-protection-threshold=0 frequency-offset=0

[admin@MikroTik] >
```

Figura 3.4: Configuración de las interfaces inalámbricas en Winbox.

Para montar el servidor se utilizó una máquina virtual Debian 6.0.7 de 32 bits corriendo bajo Oracle VM VirtualBox a la cual se le configuró una interfaz con IP



### 3.3. Herramientas utilizadas

perteneciente a la red LAN y se le instalaron los paquetes *iperf*, *snmp* y *rrdtool* que fueron necesarios para la implementación del modelo en software, el desarrollo de las pruebas y la implementación del sistema de monitoreo.

## 3.3. Herramientas utilizadas

Tanto para la configuración del AP como para el desarrollo de las pruebas y el diseño de los scripts se utilizaron diversas aplicaciones. En esta sección se presentan cada una de ellas.

### 3.3.1. Iperf

Iperf [20] es una herramienta utilizada para medir el throughput TCP máximo que se puede obtener sobre una red tanto en subida como en bajada en base a distintos parámetros o factores definidos por el usuario como ser la ventana TCP, o la existencia o no de tráfico en ambos sentidos. A su vez permite generar tráfico UDP de determinado throughput para luego medir, entre otras cosas, la pérdida de paquetes. Para este proyecto resultó de gran ayuda a la hora de la realización de pruebas preliminares de throughput.

### 3.3.2. Winbox

Winbox [21] es una aplicación desarrollada por MikroTik que permite la administración total del sistema operativo que corre sobre los equipos MikroTik (RouterOS). Presenta una interfaz relativamente sencilla e intuitiva que en definitiva hace que configurar los equipos sea mucho más fácil que conectándose directamente a ellos a través del protocolo Telnet por lo que fue de gran ayuda a la hora de la configuración de la red de laboratorio. Cuenta también con una gran cantidad de utilidades adicionales que fueron utilizadas durante distintas etapas del proyecto como por ejemplo la visualización del tráfico por sus interfaces, análisis de paquetes, o el respaldo de configuraciones.

### 3.3.3. Oracle VirtualBox VM

VirtualBox [22] es un software de virtualización multiplataforma que permite ejecutar bajo un determinado sistema operativo (en este caso Windows) múltiples máquinas virtuales que pueden correr una gran variedad de sistemas operativos. De esta manera se pueden ejecutar en un sistema operativo aplicaciones no diseñadas para él, así como albergar servidores virtuales. Esta última es la utilidad que se le da durante todo el transcurso del proyecto.

### 3.3.4. WinSCP

WinSCP [23] es un cliente SFTP, FTP, y SCP para Windows. Fue utilizado tanto para subir archivos al servidor para su posterior descarga por los distintos equipos durante las pruebas como para realizar respaldos de scripts implementados y resultados obtenidos durante el desarrollo del proyecto.

### 3.3.5. Manage Engine Free SNMP MIB Browser

Manage Engine Free SNMP MIB Browser [24] es un explorador de MIBs. Ofrece una interfaz para explorar fácilmente la estructura jerárquica de archivos MIB estandarizados o propietarios de los distintos fabricantes de equipos de telecomunicaciones. Permite también realizar consultas SNMP directamente desde la interfaz, lo cual facilitó mucho el diseño de los scripts ya que permitió determinar por cuáles OIDs era posible consultarle al AP.

### 3.3.6. RRDtool

RRDtool [25] es una herramienta de código abierto para el registro, procesamiento y análisis de datos. Es fácilmente integrable en scripts Shell, Perl y Python para realizar gráficas de datos en tiempo real, como se muestra en la figura 3.5, lo que la hace una gran herramienta para el monitoreo de redes de datos.

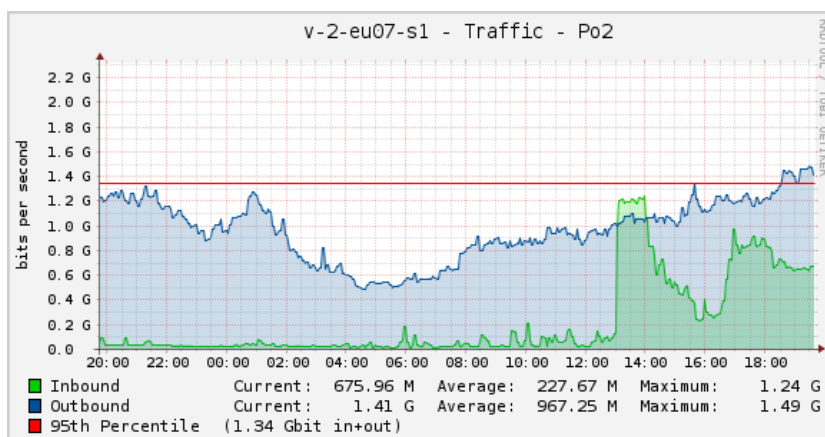


Figura 3.5: Ejemplo de gráfica generada con RRDtool: tráfico por una interfaz.

# Capítulo 4

## Primer modelo de capacidad

Habiendo ya descrito en el capítulo 2 los estándares y protocolos necesarios para el desarrollo de este proyecto y también el modelo de la red a implementar en el laboratorio, como fue detallado en el capítulo anterior, se está en condiciones de avanzar en la obtención del modelo matemático que estime la capacidad total de la red.

A continuación se presenta el modelo matemático realizado para la estimación a obtener, detallando las hipótesis y consideraciones realizadas. Existe vasta documentación acerca de la performance de redes inalámbricas en variados escenarios, utilizando distintas configuraciones y tomando diversas hipótesis. Sin embargo, aquí sólo se presentan aquellos análisis aplicables a este escenario particular.

### 4.1. Escenario e hipótesis

La situación estudiada en este primer modelo consiste en una red compuesta por un AP (Access Point), detrás del cual se encuentran un servidor, y diversos nodos que a través del AP se conectan al mismo para descargar un archivo que se considera infinitamente grande.

Este mismo escenario ha sido analizado por varios autores, de formas más o menos complejas, como puede verse en los trabajos presentados en [4], [5], [6] y [7]. Para el cumplimiento de los objetivos de este proyecto alcanza con el análisis realizado en [4]. Aquí los autores suponen en primer lugar que todos los nodos están utilizando la misma versión del estándar 802.11. Se asume además que el AP tiene un buffer infinito lo que asegura que no habrá pérdida de paquetes por desborde en su cola de transmisión.

Los autores de [4] realizan el análisis presentado a continuación.

En primer lugar, dado que se trata de tráfico de bajada, podría pensarse que

## Capítulo 4. Primer modelo de capacidad

todos los paquetes transmitidos son enviados por el AP, en cuyo caso no habrían colisiones por ser este el único nodo que estaría compitiendo por el acceso al medio. Sin embargo, recordando lo visto en la sección 2.1.1, esto no es cierto debido a que las estaciones deben enviar un paquete de reconocimiento tras cada segmento TCP recibido. Es necesario entonces hacer un análisis más profundo de la situación para determinar qué sucede con las colisiones de paquetes.

Dado que no hay pérdida de paquetes en el AP y suponiendo que la probabilidad de que una estación alcance el límite de retransmisiones de la MAC es muy baja, se puede asumir que el AP siempre tiene un paquete disponible para enviar. En segundo lugar, dado que las estaciones únicamente envían reconocimientos al AP, sólo aumentarán la ocupación de su buffer tras recibir un paquete, es decir cuando el AP logró acceder al medio para enviarles un paquete y no colisionó con ninguna otra estación. Dado que DCF otorga igual posibilidad de transmisión a todos los nodos que estén compitiendo por el medio, la probabilidad de que el AP envíe un paquete a un nuevo destino, aumentando así la cantidad de estaciones que compiten por el medio, es muy baja. Esto permite concluir que el número de estaciones que compiten por enviar reconocimientos se mantendrá bajo, lo que tendrá como consecuencia un throughput relativamente alto y una baja probabilidad de colisiones.

Es interesante destacar que, a pesar de que se incremente el número de estaciones conectadas en la red, no solamente será alto el throughput y bajas las colisiones, sino que también estos valores se mantendrán relativamente constantes. Esto permite concluir que el rendimiento de TCP sobre redes WLAN no se ve degradado con el número de estaciones; el throughput por estación escala como  $1/n$ , siendo  $n$  la cantidad de nodos conectados a la red.

### 4.2. Obtención del modelo

Realizando un análisis de los tiempos involucrados en la transmisión de una trama, de acuerdo a lo visto en la sección 2.1.5, puede obtenerse un modelo que aproxime el throughput que debería obtener un nodo, cursando tanto tráfico down-link como le es posible, en base a sus tasas de conexión y a diversos parámetros propios de la revisión del estándar utilizada. Luego, utilizando dicho resultado, puede obtenerse un modelo que aproxime el throughput total que se espera obtengan todos los usuarios conectados a la red.

Por cada paquete enviado por el AP, la estación que lo recibe debe enviar un mensaje de reconocimiento  $TCP_{ack}$ , por lo que, al cabo de un ciclo, el tiempo total que se necesita para enviar la carga útil debe tomar en consideración tanto el tiempo de transmisión del paquete de datos como el tiempo de transmisión del reconocimiento. Como se está considerando un escenario en el que todos están descargando un archivo infinitamente largo, el AP siempre tendrá un paquete listo

## 4.2. Obtención del modelo

para enviar. Cada vez que envíe un nuevo paquete comenzará la cuenta regresiva establecida por el mecanismo CSMA/CA. Asumiendo que no hay colisiones, el tiempo medio de cada cuenta regresiva es  $(W_{min} - 1)/2$  slots de tiempo. Debido a la simetría del escenario estudiado, la estación destino de los paquetes enviados puede ser cualquiera de las estaciones de manera equiprobable. Además, como se vió en la sección anterior, son pocas las estaciones que están sensando el canal con paquetes para transmitir, por lo que es muy alta la probabilidad de que la estación que recibe un paquete detecte que el canal está libre y pueda enviar el  $TCP_{ack}$  de forma inmediata, sin necesidad de realizar nuevamente el proceso de conteo regresivo.

De lo anterior, se obtiene que el throughput que obtendría cada nodo si estuviera solamente él conectado a la red sería aproximadamente:

$$C_i = \frac{E}{aSlotTime \cdot \frac{(W_{min} - 1)}{2} + T_{data} + T_{ack}}, \quad (4.1)$$

donde  $T_{data} + T_{ack}$  es el tiempo total que le lleva al nodo  $i$  descargar un paquete completo de carga útil  $E$ . Dado que se está trabajando bajo la hipótesis de que los nodos están descargando un archivo muy grande,  $E$  toma el valor de 1460 bytes<sup>1</sup>. Aquí se desprecian los tiempos de propagación de los paquetes en el aire que suelen estar en el entorno de los 10 ns. Según lo visto en la sección 2.1.5, los tiempos  $T_{data}$  y  $T_{ack}$  pueden descomponerse de la siguiente manera:

$$T_{data} = DIFS + SIFS + T_{802,11-data} + T_{802,11-ack}$$

$$T_{ack} = DIFS + SIFS + T_{tcp-ack} + T_{802,11-ack}$$

donde para 802.11b se tiene:

$$T_{802,11-data} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{802,11_{data}}{Tx_{rate}}$$

$$T_{802,11-ack} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{802,11_{ack}}{B_{rate}}$$

$$T_{tcp-ack} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{TCP_{ack}}{Rx_{rate}}$$

y para 802.11g se tiene:

$$T_{802,11-data} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{802,11_{data}}{Tx_{rate}} + T_{signal-extension}$$

$$T_{802,11-ack} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{802,11_{ack}}{B_{rate}} + T_{signal-extension}$$

<sup>1</sup>Usualmente  $MTU_{ethernet} = 1500 = 1460 + 20 + 20$  bytes correspondientes a carga útil (payload), encabezado IP y encabezado TCP respectivamente.

## Capítulo 4. Primer modelo de capacidad

$$T_{tcp-ack} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{TCP_{ack}}{R_{xrate}} + T_{signal-extension}$$

Aquí  $B_{rate}$  es la tasa básica (o basic rate) configurada en el AP (ver sección 3.2),  $PLCP_{header}$  es el tamaño del encabezado de PLCP,  $PLCP_{preamble}$  es el tamaño del preámbulo de PLCP,  $R_{xrate}$  es la tasa de recepción del AP para ese nodo,  $T_{xrate}$  es la tasa de transmisión del AP para ese nodo, y  $T_{signal-extension}$  es un tiempo adicional de  $6 \mu s$  que debe esperarse según la sección 19.3.2.3 de [26] para asegurar que el proceso de decodificación puede llevarse a cabo correctamente. Los valores para todos estos parámetros pueden verse en las tablas 4.1 y 4.2.

Parámetro	802.11b	802.11g
$802,11_{ack}$ (bits)	112	216 <sup>2</sup>
$802,11_{data}$ (Bytes)	1536	1536
$TCP_{ack}$ (Bytes)	76	76
$PLCP_{header}$ (bits)	48	48
$PLCP_{preamble}$ (bits)	144	72

Tabla 4.1: Parámetros de los estándares 802.11b y 802.11g

Parámetro	802.11b	802.11g <sup>3</sup>
aSlotTime ( $\mu s$ )	20	9
SIFS ( $\mu s$ )	10	10
DIFS ( $\mu s$ )	50	28 o 50
EIFS ( $\mu s$ )	364	160
$T_{signal-extension}$ ( $\mu s$ )	-	6
$W_{min}$	31	15

Tabla 4.2: Parámetros de los estándares 802.11b y 802.11g

Luego se calcula la capacidad total de la red como la relación entre los datos útiles traficados por todos los usuarios y el tiempo total que les lleva hacerlo:

$$C = \frac{n \cdot E}{\sum_{i=1}^n T_i}, \quad (4.2)$$

<sup>2</sup>Son 112 bits pero la mínima unidad que se puede transmitir es un símbolo (216 bits)

<sup>3</sup>Se trabajó con 802.11 only g con preámbulo corto

### 4.3. Implementación del modelo

donde  $n$  es la cantidad de usuarios conectados y  $T_i$  es el tiempo medio que demora cada uno en traficar  $E$  bytes de carga útil que es igual a:

$$T_i = aSlotTime \cdot \frac{(W_{min} - 1)}{2} + T_{data} + T_{ack},$$

Reescribiendo la ecuación 4.2 se obtiene,

$$C = \frac{1}{\frac{1}{n} \cdot \sum_{i=1}^n \frac{T_i}{E}},$$

que es equivalente a

$$C = \frac{1}{\sum_{i=1}^n \frac{1}{n} \frac{1}{C_i}} \quad (4.3)$$

Este último resultado representa la capacidad total de la red suponiendo que todas los usuarios conectadas están traficando lo máximo posible.

### 4.3. Implementación del modelo

Para la implementación del modelo 1 se desarrollaron tres scripts en lenguaje Shell que deben ser ejecutados en el servidor Debian que se encuentra detrás del AP. Se decidió utilizar dicho lenguaje de scripting por conocimientos previos que se tenían de él y por considerarlo más que suficiente para la obtención y el procesamiento de las medidas obtenidas.

El primero de estos scripts, *consultaSNMP\_rate.sh*, se encarga de la medición del throughput real de todos los usuarios de la red y su comparación con el valor esperado según el modelo 1. Es utilizado para realizar pruebas puntuales. El segundo script, *llamo\_consultaSNMP\_rate.sh* fue desarrollado para realizar pruebas de forma sucesiva y guardar los resultados detallando la cantidad de medidas de throughput tomadas y bajo qué condiciones fueron realizadas. Finalmente, el tercer script implementado, *throughputs\_time.sh* fue utilizado para determinar el tiempo óptimo de las pruebas de validación.

El código de cada uno de estos scripts puede ser consultado en el CD adjunto en esta documentación.

## consultaSNMP\_rate.sh

El script *consultaSNMP\_rate.sh* tiene como entradas a ingresar por el usuario la duración total de la prueba, la revisión de 802.11 configurada en el AP (b o g) y el tiempo que se debe esperar entre consultas al AP. Su salida es el throughput total medido y la estimación obtenida por el modelo 1. A su vez devuelve como resultados adicionales las direcciones MAC de cada nodo y el throughput correspondiente a cada uno de ellos. Despliega también un mensaje indicando si durante el transcurso de la prueba se conectó algún usuario nuevo a la red, en cuyo caso la prueba es descartada. Debido a que este modelo será validado únicamente en entorno de laboratorio, se asume que no habrán desconexiones de usuarios durante el transcurso de las pruebas. El funcionamiento del script se esquematiza en la figura 4.1.

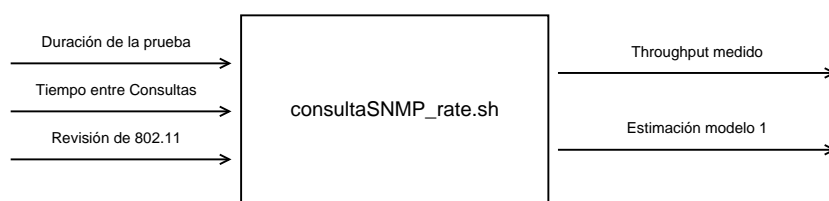


Figura 4.1: Diagrama del script *consultaSNMP\_rate.sh*

El script se estructura en tres grandes bloques funcionales. El primero de ellos consiste en interrogar al AP mediante diversas consultas SNMP y guardar los datos obtenidos para cada usuario de manera ordenada. El segundo bloque calcula, a partir de la información recolectada, el throughput obtenido por cada uno de los nodos y el throughput conjunto. Finalmente, el tercer bloque realiza, a partir de la información recolectada con las consultas SNMP, una estimación del throughput total de la red según el modelo 1 obtenido en la sección anterior.

La precisión de esta estimación depende fuertemente de las tasas de conexión de transmisión y recepción de los nodos, por lo que, recordando que estas suelen variar mucho con el tiempo, no sería adecuado interrogar al AP por las tasas de conexión únicamente al principio o al final de la prueba y asumir que se mantienen constantes durante el transcurso de esta. Por lo tanto, para minimizar el impacto de dichas variaciones en la estimación de throughput, se definen varios intervalos intermedios en los que se consultan las tasas de conexión de cada nodo en base a lo cual se calcula la estimación de throughput del modelo 1 para el conjunto de los nodos. Luego se calcula un promedio de las estimaciones obtenidas a lo largo de los distintos intervalos siendo este el throughput esperado de la red.

El tiempo que se espera entre las interrogaciones es especificado por el usuario al invocar el script e idealmente debe ser pequeño de tal forma de detectar la mayor cantidad posible de cambios en las tasas de conexión. Este tiempo junto con el tiempo de duración de la prueba determinan la cantidad de intervalos de



### 4.3. Implementación del modelo

interrogación. Cada intervalo quedará definido por una consulta SNMP inicial y una final. Cabe aclarar que, a los efectos de calcular la estimación de throughput para cada nodo en cada uno de estos pequeños intervalos, se supone que dicho nodo se mantuvo conectado durante el intervalo a la tasa obtenida en la consulta final. Los resultados con esa hipótesis deberían ser los mismos si se supusiese que dicho nodo se mantiene conectado a la tasa que tiene en la primera consulta.

Durante la consulta inicial, lo que se hace en primer lugar es averiguar la cantidad de nodos conectados al AP. Para ello se hace una consulta SNMP al AP por el OID correspondiente a:

- Wireless Registration Table Entry Count<sup>4</sup>

Una vez obtenido dicho valor se consulta por la dirección MAC de cada uno de los usuarios conectados y la cantidad de bytes transmitidos por el AP hacia cada uno de ellos. En otras palabras, se realizan consultas SNMP por los OIDs correspondientes a:

- Wireless Registration Table Address<sup>5</sup>
- Wireless Registration Table Transmitted Bytes<sup>6</sup>

Esto permite asociar a cada nodo la cantidad de bytes que llevaba cursados al inicio de la prueba.

Luego, en las consultas intermedias únicamente se interroga al AP por las direcciones MAC de los nodos conectados en dicho instante y sus tasas de conexión. En estas no se interroga la cantidad de bytes transmitidos a cada nodo ya que el throughput real es medido en base a la diferencia entre los bytes cursados en la última y en la primera consulta y al tiempo de ejecución de la prueba. Es decir, en las consultas intermedias se interroga al Access Point únicamente por los OIDs correspondientes a:

- Wireless Registration Table Address<sup>7</sup>
- Wireless Registration Table Reception Rate<sup>8</sup>
- Wireless Registration Table Transmission Rate<sup>9</sup>

La última consulta es idéntica a la primera, con la salvedad de que en este caso se agrega la consulta de las tasas de recepción y transmisión de los nodos tal como se hace en las consultas intermedias de manera de contar con los datos suficientes

---

<sup>4</sup>mtxrWIRtabEntryCount - OID .1.3.6.1.4.1.14988.1.1.1.4

<sup>5</sup>mtxrWIRtabAddr - OID .1.3.6.1.4.1.14988.1.1.1.2.1.1

<sup>6</sup>mtxrWIRtabTxBytes - OID .1.3.6.1.4.1.14988.1.1.1.2.1.4

<sup>7</sup>mtxrWIRtabAddr - OID .1.3.6.1.4.1.14988.1.1.1.2.1.1

<sup>8</sup>mtxrWIRtabRxRrate - OID .1.3.6.1.4.1.14988.1.1.1.2.1.9

<sup>9</sup>mtxrWIRtabTxRrate - OID .1.3.6.1.4.1.14988.1.1.1.2.1.8

## Capítulo 4. Primer modelo de capacidad

para calcular la estimación de throughput para cada nodo en el último intervalo.

Luego de finalizadas todas las consultas, en el segundo bloque del script, se calcula el throughput total medido para cada nodo en base a los datos de los bytes transmitidos en la primera y en la última consulta de la siguiente manera:

$$\text{Throughput} = \frac{\text{Bytes}(t_f) - \text{Bytes}(t_0)}{t_f - t_0}$$

donde  $t_0$  es el tiempo de inicio de la prueba (en segundos),  $t_f$  es el tiempo final de la prueba (en segundos),  $\text{Bytes}(t_0)$  son los bytes que ha descargado el nodo hasta el inicio de la prueba y  $\text{Bytes}(t_f)$  son los bytes que ha descargado el nodo hasta el final de la prueba. Nótese que el tiempo  $t_f - t_0$  difiere del tiempo de duración de prueba que ingresó originalmente el usuario ya que la ejecución de las consultas SNMP intermedias añade ciertas demoras al tiempo ingresado por el usuario.

Luego, a partir de los throughputs individuales de cada nodo se calcula también el throughput total de la red como la suma de todos estos.

Finalmente, en lo que sería el tercer bloque del script, se aplica la ecuación del modelo 1 (ecuación 4.3) a cada uno de los pequeños intervalos con lo cual se obtiene el throughput esperado en cada uno de ellos. Luego, haciendo un promedio temporal de los throughputs de cada intervalo se obtiene la estimación de cuánto throughput deberían obtener entre todos los usuarios descargando simultáneamente un archivo infinitamente grande como se muestra en la siguiente ecuación:

$$C_{esperada} = \frac{\sum_{i=1}^k C_i}{k}$$

donde  $C_i$  es el throughput total esperado para el intervalo  $i$  y  $k$  es la cantidad total de intervalos.

De esta manera, gracias a haber consultado las tasas de conexión de cada nodo en cada intervalo, el valor de la estimación dependerá de la variación de estos.

La salida del script `consultaSNMP_rate.sh` para una prueba con 8 usuarios conectados es de la siguiente forma:

### 4.3. Implementación del modelo

```
===== 03/03/15 21:49 =====
Tiempo de ejecución: 62.091525625 segundos

User 1 MAC: 00 21 85 F5 7A 1E --- Rate: 1586.85 Kbps
User 2 MAC: 00 23 15 4D 1F FC --- Rate: 1596.27 Kbps
User 3 MAC: 00 24 21 4F 83 7C --- Rate: 1339.77 Kbps
User 4 MAC: 78 92 9C A4 76 3C --- Rate: 1581.05 Kbps
User 5 MAC: 78 92 9C A8 63 84 --- Rate: 1634.24 Kbps
User 6 MAC: 8C 3A E3 95 B3 B4 --- Rate: 1512.61 Kbps
User 7 MAC: 90 00 DB E2 A2 64 --- Rate: 1621.95 Kbps
User 8 MAC: 9C E7 OD 5C A5 E9 --- Rate: 1558.22 Kbps

Throughput Total Medido: 12430.96 Kbps
Throughput Total Esperado: 12447.65 Kbps

Se conectaron 0 usuarios durante el transcurso de la prueba.
```

#### llamo\_consultaSNMP\_rate.sh

El script *llamo\_consultaSNMP\_rate.sh* permite realizar, invocando al script *consultaSNMP\_rate.sh*, tantas mediciones de throughput como se quiera de forma sucesiva. Fue fundamental para realizar en el menor tiempo posible todas las mediciones necesarias para validar el modelo de estimación. Tiene como entrada adicional la cantidad de mediciones que se desea realizar como se muestra en el diagrama de la figura 4.2.

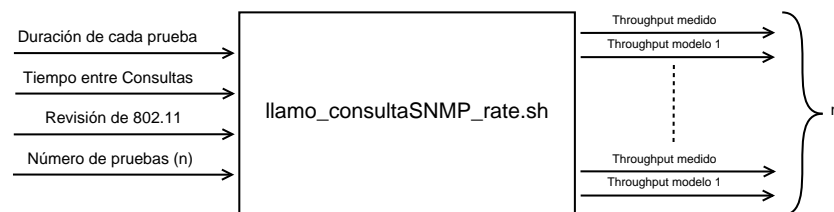


Figura 4.2: Diagrama del script *llamo\_consultaSNMP\_rate.sh*

#### throughputs\_time.sh

Adicionalmente se implementó un script para determinar el tiempo óptimo de duración de las pruebas para la validación del modelo. Este script, *throughputs\_time.sh*, también implementado en Shell, determina en una sola ejecución el throughput total promedio obtenido por los equipos de la red en los primeros 10, 20, 30,..., y 120 segundos tras su ejecución.

Esto se realiza consultando en primera instancia al AP por la cantidad de usuarios conectados, sus direcciones MAC, y la cantidad de bytes que han descargado hasta el momento. Luego, periódicamente cada 10 segundos, se consulta nuevamente por la dirección MAC y los bytes descargados de todos los equipos y se calcula el throughput obtenido para cada uno de ellos como la diferencia entre los bytes descargados entre el tiempo inicial y cada tiempo intermedio (hasta llegar a los 120 segundos).

## 4.4. Elección del tiempo de medición

Antes de comenzar con las pruebas de validación del modelo 1, es necesario determinar el tiempo óptimo de duración de las mediciones. Este tiempo debe ser suficientemente largo como para permitir cierta estabilización en las velocidades de descarga y a su vez no tan largo para evitar, en la medida de lo posible, la aparición de problemas tales como pérdidas o desconexiones temporales de usuarios.

Para la elección de dicho tiempo se realizan varias pruebas utilizando el script *throughputs\_time.sh* que devuelve el throughput total obtenido por los equipos de la red en distintos intervalos de medición. Luego se calcula la desviación estándar entre los resultados obtenidos para cada uno de los intervalos.

En la figuras 4.3 y 4.4 pueden verse tanto los resultados de throughput medido tras la ejecución del script como también la desviación estándar de los resultados de las 10 pruebas para cada uno de los intervalos de tiempo.

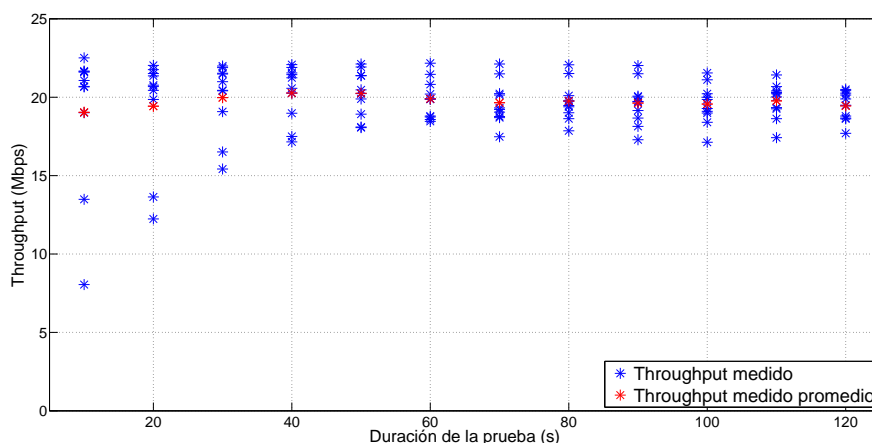


Figura 4.3: Throughput de la red según mediciones de distinta duración.

Puede observarse que cuando se calcula el throughput en intervalos chicos de tiempo la desviación estándar es muy alta. Este es un resultado esperable debido a que disminuciones en las tasas de conexión se pagan más caro cuando el intervalo de tiempo es menor. Es decir, la mayor parte del tiempo los nodos se encuentran conectados a tasas altas, por lo cual una disminución de esta tendrá más impacto en el resultado final cuando se trata de un intervalo de 10 o 20 segundos que cuando se trata de un intervalo de 120 segundos.

Por otra parte se puede apreciar que en general a partir de los intervalos de tiempo mayores o iguales a 60 segundos de duración la desviación estándar de las medidas no toma valores excesivamente grandes.

## 4.5. Pruebas y validación del modelo

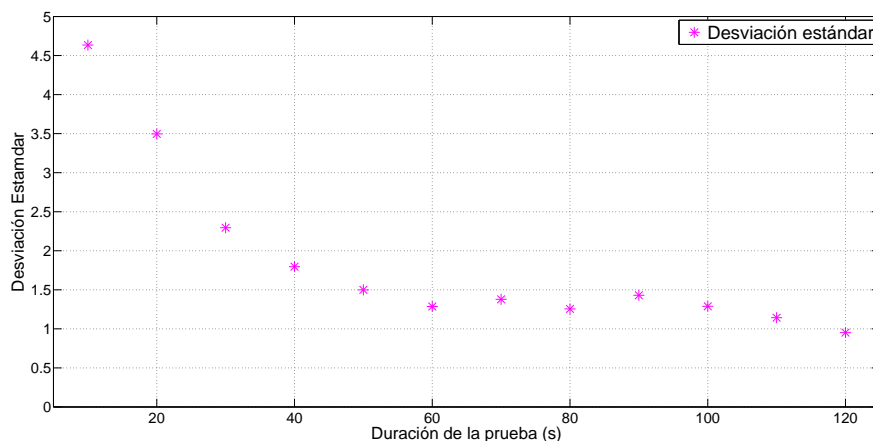


Figura 4.4: Desviación estándar del throughput para cada intervalo de medición.

Por lo anterior se elige una duración de 60 segundos para las pruebas de validación del modelo 1.

## 4.5. Pruebas y validación del modelo

Para la realización de las pruebas se dispone en el servidor de un repositorio de archivos en donde se aloja un fichero de aproximadamente 1 GB de tamaño que, a los efectos de la duración de las pruebas y de las velocidades de descarga de los nodos, puede considerarse infinitamente grande. A continuación se conecta una laptop y comienza la descarga de dicho fichero, y con la ayuda del script *llamo\_consultaSNMP\_rate.sh* se realizan 10 estimaciones (y mediciones) sucesivas de throughput. Se repite el proceso agregando una a una las laptops y obteniendo la estimación de throughput mediante la ejecución de dicho script para cada número de nodos conectados.

El primer hecho que se observa es la uniformidad con que se distribuye el tráfico entre los usuarios conectados, lo que era esperado a raíz de los resultados presentados en [4] y de la simetría del escenario y como consecuencia de la equiprobabilidad de acceso al medio otorgada por CSMA/CA. Esto se puede ver en la figura 4.5 donde se muestra la distribución de tráfico en una prueba con 3 usuarios conectados.

En las figuras 4.6 y 4.7 se puede observar cómo evolucionan las tasas de conexión para cada usuario a lo largo del tiempo. Como ya se ha dicho, las variaciones en las tasas de conexión son determinantes para la estimación de capacidad del modelo 1. El hecho de que estas varíen tanto, no solo justifica, sino que hizo indispensable la implementación de un script que consulte periódicamente por dichas tasas de manera de obtener un resultado más confiable.

## Capítulo 4. Primer modelo de capacidad

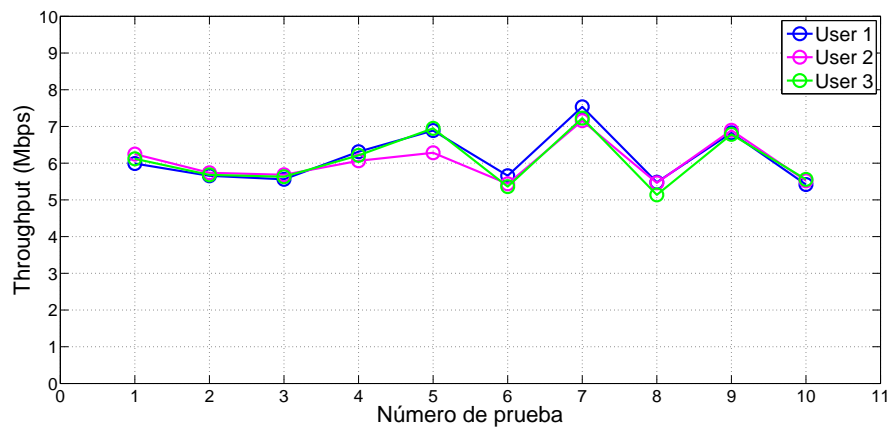


Figura 4.5: Reparto de throughput para 3 usuarios conectados.

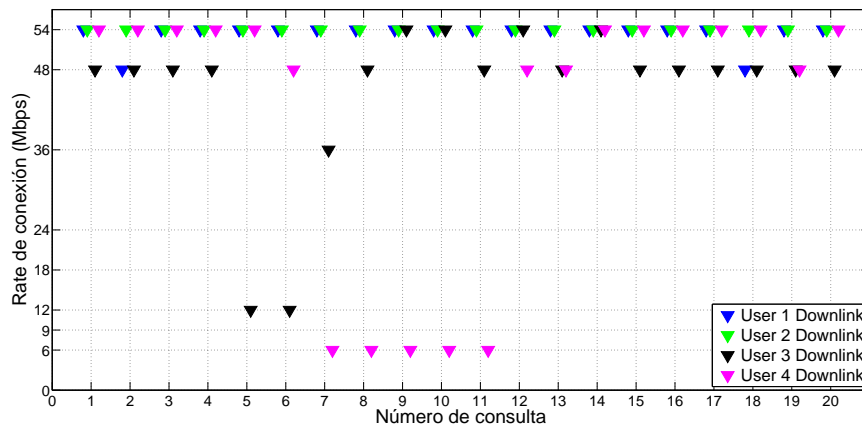


Figura 4.6: Tasas de bajada en una prueba de laboratorio con 4 usuarios conectados.

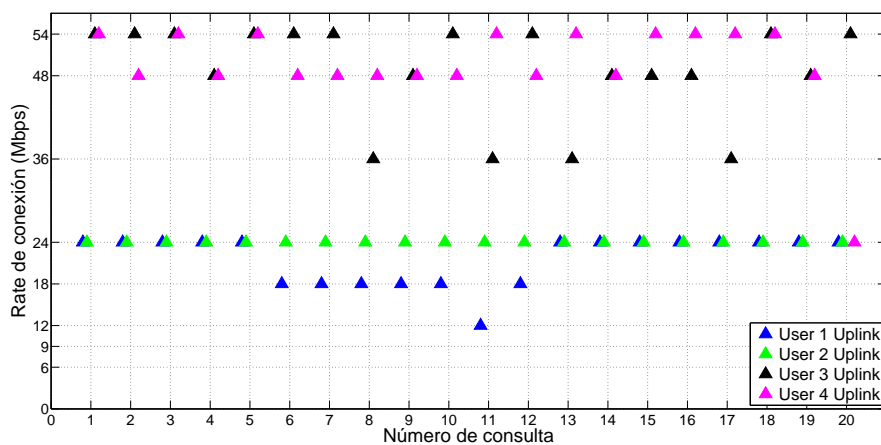


Figura 4.7: Tasas de subida en una prueba de laboratorio con 4 usuarios conectados.

## 4.5. Pruebas y validación del modelo

Se observa además, como se muestra en la figura 4.8, que el throughput estimado es prácticamente invariante ante grandes cambios en las tasas de subida, mientras que es muy sensible a cambios en las tasas de bajada. Se puede apreciar que para las primeras tres consultas realizadas, en las cuales se tienen tasas de conexión altas, el throughput ronda los 20 Mbps. Luego, en las consultas 5 y 6, donde hay una gran caída en la tasa de bajada, también se percibe una gran caída de throughput a valores que rondan los 9 Mbps. Finalmente, se puede apreciar en las consultas 8, 11, 13 y 17, que a pesar de tener una tasa de subida considerablemente menor que la de las consultas iniciales, el hecho de tener una alta tasa de bajada hace que se obtenga un throughput muy cercano al máximo de 20 Mbps.

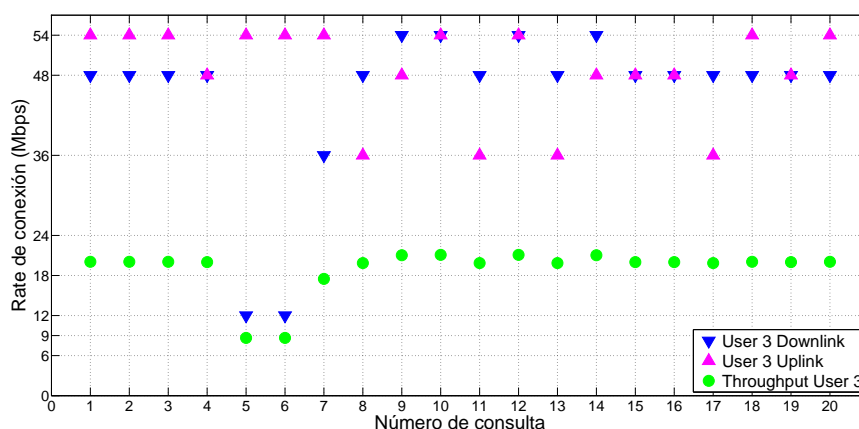


Figura 4.8: Variación de tasas de conexión de bajada y subida y throughput estimado para un usuario particular.

De la figura 4.9 a la figura 4.17 se presentan los resultados obtenidos tras las pruebas variando entre 1 y 9 la cantidad de usuarios descargando un archivo de la red. En estas se puede apreciar el buen comportamiento del modelo 1 implementado ya que el throughput estimado sigue muy de cerca en todos los casos el comportamiento del throughput medido.

Para el análisis de los resultados se promedian los throughputs medidos y esperados para las 10 pruebas realizadas para las distintas cantidades de usuarios conectados. Estos valores se muestran en la figura 4.18. Para dar por validado el modelo 1 es necesario obtener valores concretos que cuantifiquen su precisión. Para tal propósito se calcula el error relativo entre los valores estimados y medidos.

Como puede verse en la figura 4.19 el error relativo entre el throughput medido y el esperado se encuentra siempre por debajo del 7%, siendo incluso menor al 4% para los casos en que hay más de dos usuarios conectados. Dadas las grandes variaciones de performance que pueden sufrir las redes inalámbricas debido a una gran variedad de factores, un error de esta magnitud en la estimación se considera más que aceptable.

## Capítulo 4. Primer modelo de capacidad

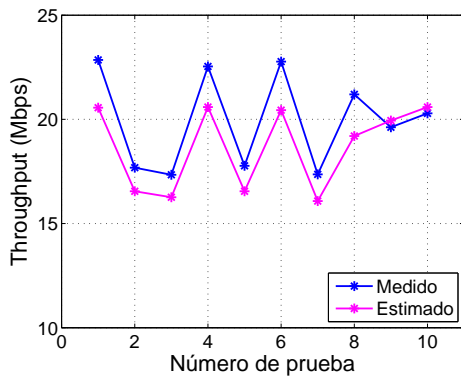


Figura 4.9: Resultados modelo 1: 1 usuario.

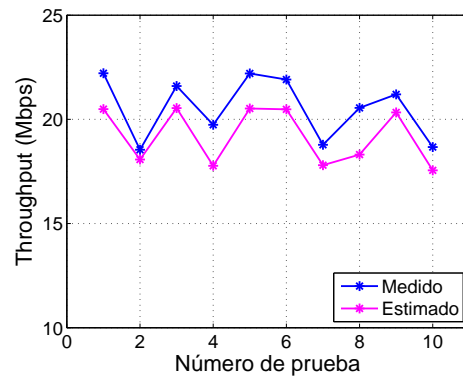


Figura 4.10: Resultados modelo 1: 2 usuarios.

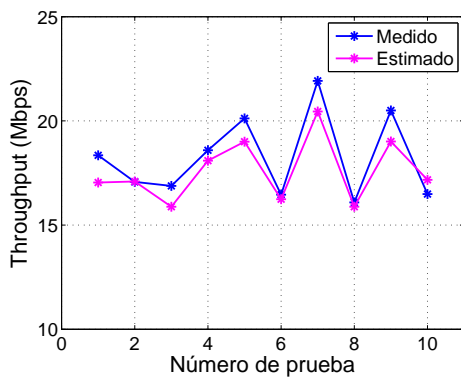


Figura 4.11: Resultados modelo 1: 3 usuarios.

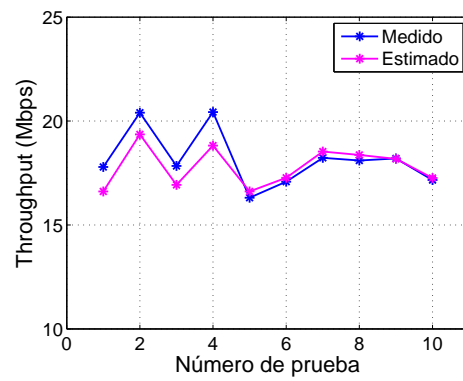


Figura 4.12: Resultados modelo 1: 4 usuarios.

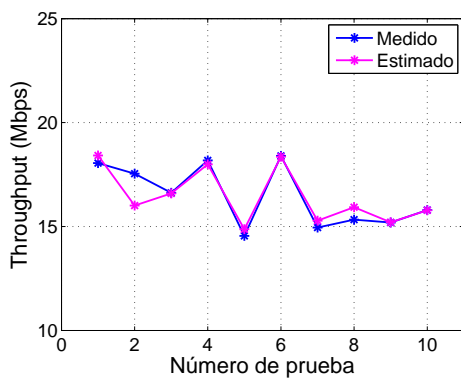


Figura 4.13: Resultados modelo 1: 5 usuarios.

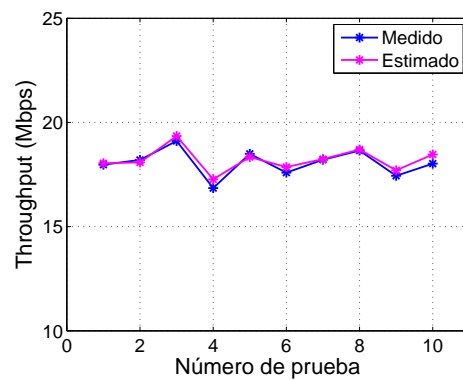


Figura 4.14: Resultados modelo 1: 6 usuarios.



#### 4.5. Pruebas y validación del modelo

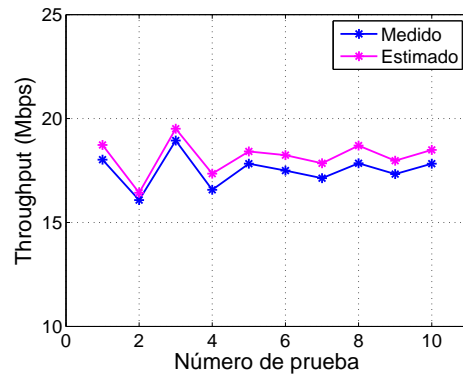
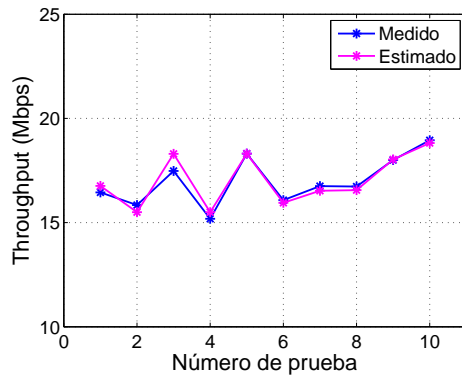


Figura 4.15: Resultados modelo 1: 7 usuarios. Figura 4.16: Resultados modelo 1: 8 usuarios.

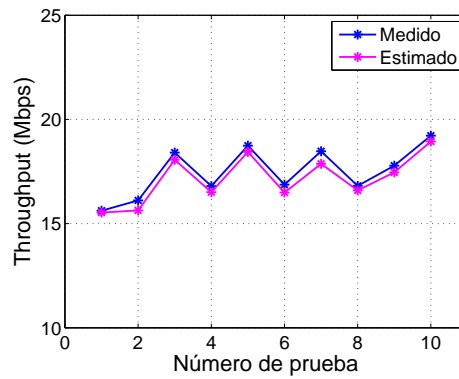


Figura 4.17: Resultados modelo 1: 9 usuarios.

Estos resultados cumplen con el objetivo planteado (errores relativos menores a 10

## Capítulo 4. Primer modelo de capacidad

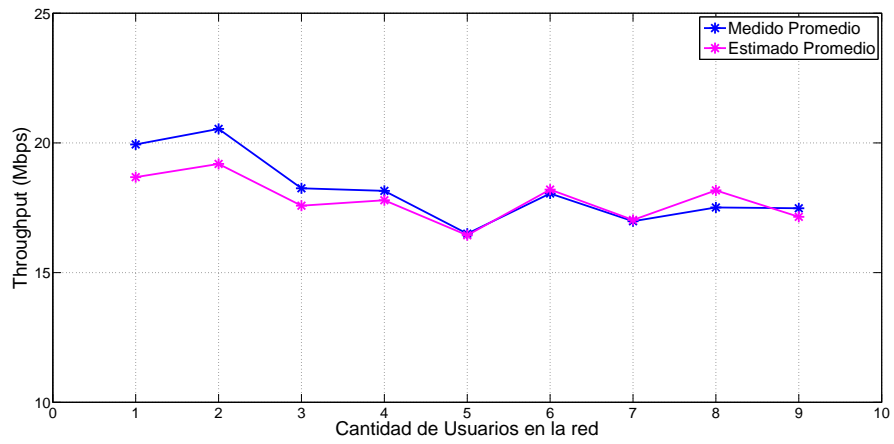


Figura 4.18: Throughput promedio medido y estimado para distintas cantidades de usuarios conectados.

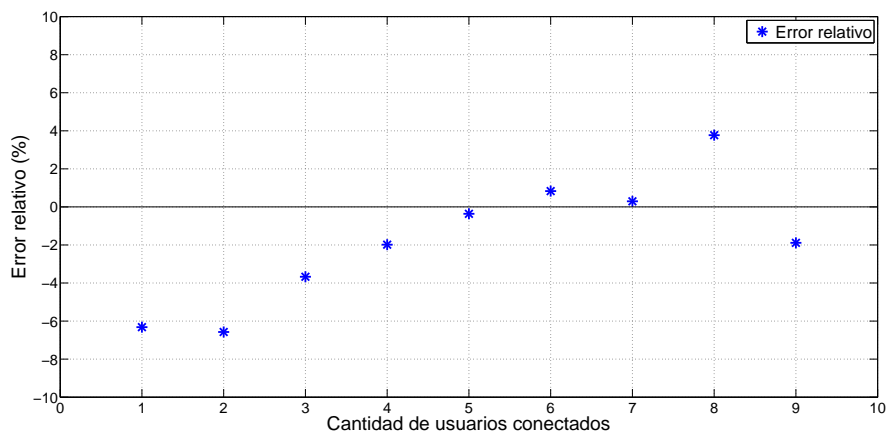


Figura 4.19: Error relativo del throughput estimado promedio respecto al medido promedio.

# Capítulo 5

## Segundo modelo de capacidad

Luego de haber obtenido y validado en el capítulo anterior un modelo matemático capaz de estimar cuál es la capacidad total de la red en función de las características de los usuarios conectados, se pretende ahora obtener un nuevo modelo que estime el throughput máximo que puede obtener un nuevo usuario que se conecte a la red. En esta nueva situación el estado de la red previo a su entrada no solamente es desconocido y puede ser cualquiera, sino que también puede ser muy variable con el tiempo debido a conexiones o desconexiones de usuarios, o variaciones en el tráfico demandado por ellos. Es esperable que este hecho agregue dificultades a la hora de la validación del modelo principalmente fuera del laboratorio, por lo cual, como se verá más adelante, habrá que tomar ciertas hipótesis con respecto a la variación del tráfico para su validez.

En este capítulo se describe el escenario planteado para este nuevo modelo, las hipótesis consideradas, el cálculo de algunos modelos matemáticos, su implementación en software, y las pruebas y validación de dichos modelos. Se presentan también los resultados de las pruebas obtenidas con cada modelo en la red inalámbrica de Plan Ceibal de una escuela pública como forma de validación del modelo fuera del entorno de laboratorio.

### 5.1. Escenario e hipótesis

En la red considerada para este escenario puede existir tanto tráfico TCP como UDP en ambos sentidos y se desconoce la cantidad de nodos conectados y su comportamiento. A su vez, de la misma forma en que se hizo en el modelo 1, la implementación de este nuevo modelo de capacidad se hará mediante consultas SNMP al AP. El hecho de que la MIB de MikroTik no permita interrogar al AP por los dos tipos de tráfico de forma independiente, hace que este modelo no pueda distinguir entre tráfico TCP y UDP y todos los cálculos se realizarán asumiendo que el tráfico cursado es únicamente TCP. Cabe recordar que este proyecto está orientado a redes inalámbricas del Plan Ceibal, en las cuales considerar que

## Capítulo 5. Segundo modelo de capacidad

todo el tráfico cursado es TCP es una hipótesis razonable.

Como se mencionó anteriormente, el estado de la red en este nuevo escenario puede variar significativamente de un momento a otro con lo cual también varía la capacidad total real de la red. Debido a que la estimación para la capacidad futura de un nuevo usuario se basa en estados pasados de la red, el modelo sólo podrá ser validado cuando su estado no sufre grandes cambios en pequeños intervalos de tiempo.

### 5.2. Obtención del modelo

Para realizar la estimación, lo primero que se hace es aplicar el estimador del modelo 1 a partir del cual se determina en qué escenario se encuentra la red. Se considerarán dos posibles escenarios que se diferenciarán entre sí según cuánto estén descargando los usuarios conectados. Haciendo un abuso del lenguaje, de ahora en más se referirá a los usuarios para los cuales el AP siempre tiene un paquete en cola como usuarios saturados o saturando, y al resto como usuarios no saturados. Debido a que la capacidad libre de la red dependerá fuertemente de la presencia o no de usuarios saturados, resulta conveniente distinguir entre los siguientes escenarios posibles:

- Escenario 1: No hay usuarios saturados.
- Escenario 2: Hay al menos un usuario saturado.

Antes de proceder a estimar el throughput máximo para un nuevo usuario para cualquiera de los escenarios anteriores es necesario detectar si hay usuarios que están conectados a la red pero no tienen ninguna actividad (de ahora en más serán llamados usuarios inactivos). Estos no compiten por el acceso al canal y por lo tanto no afectan la capacidad total de la red. Como el cálculo de estimación de capacidad del modelo 1 se basa en las tasas de los usuarios conectados, para evitar introducir su aporte en el cálculo, es necesario descartar los usuarios inactivos antes de realizar la estimación de capacidad. Luego de una serie de pruebas se observa que el tráfico era muy cercano a 0 Kbps o al menos del orden de decenas de Kbps. Se elige entonces un umbral de tráfico muy pequeño (5 Kbps) de manera de descartar en el cálculo a todos los usuarios inactivos.

Para realizar la discriminación entre usuarios saturados y no saturados, se realizan una serie de iteraciones mediante las que se calcula un umbral de saturación, tal que si el throughput obtenido por un usuario está por encima de este, el usuario será considerado saturado. Estas iteraciones se ejemplifican en la figura 5.1.

Para definir el umbral de saturación se calcula en primera instancia la capacidad total de la red incluyendo al nuevo usuario. Para esto se asume que el nuevo usuario se comportará de forma similar al resto por lo cual tendrá también una

## 5.2. Obtención del modelo

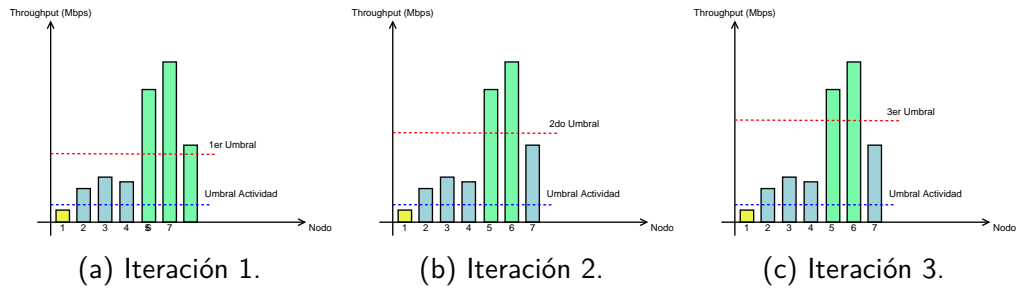


Figura 5.1: Proceso de iteración para el cálculo del umbral de saturación. En verde se indican los usuarios clasificados como saturados en cada iteración.

tasa similar, lo que en definitiva implica que el valor de la capacidad total no se verá modificado. Esto es en principio razonable pues en redes de Plan Ceibal se espera que todos los usuarios tengan dispositivos con las mismas características y estén aproximadamente a la misma distancia del AP.

De esta forma para calcular la capacidad total de la red teniendo en cuenta al nuevo usuario se aplica el modelo 1 a la totalidad de los usuarios que estaban conectados. Este valor luego se divide entre la cantidad de usuarios (incluyendo al nuevo), obteniendo el throughput que obtendría cada usuario si estuvieran todos descargando tanto como puedan. Este será el primer valor del umbral de saturación (ver figura 5.1a).

En el caso en que hay muchos usuarios conectados a la red el primer umbral de saturación tiende a ser bajo, por lo que podría suceder que un usuario que no sature la celda sea clasificado como saturado según el umbral anterior. Para evitar esto se realizan varias iteraciones para clasificar a los usuarios.

Con el primer umbral calculado se determinan los primeros candidatos a usuarios saturados. Luego se calcula la capacidad total de la red en función de dichos candidatos y el nuevo usuario, y se le resta el throughput ya consumido por los que se sabe no son saturados. Luego, al dividir el valor obtenido entre todos los candidatos y el nuevo usuario se obtiene un nuevo umbral de saturación (ver figura 5.1b) a partir del cual se descartan los candidatos cuyo throughput quede por debajo de él. Esto se repite hasta no descartar más usuarios, determinando así los usuarios saturados (ver figura 5.1c).

### 5.2.1. Escenario 1: No hay usuarios saturados.

En este escenario se distinguirán tres casos según haya:

- **Caso 1:** Tanto usuarios inactivos como activos.
- **Caso 2:** Únicamente usuarios inactivos.

## Capítulo 5. Segundo modelo de capacidad

- **Caso 3:** Ningún usuario conectado.

En el primer caso se asume que el nuevo usuario podrá obtener la capacidad total de la red excepto lo que ya está siendo consumido por quienes están conectados. Entonces la estimación de throughput para un nuevo usuario,  $C_{nuevoUsuario}$ , será:

$$C_{nuevoUsuario} = C_{total} - C_{cursado}$$

donde  $C_{total}$  es la capacidad total de la red obtenida mediante el modelo 1 (descartando previamente a los usuarios inactivos), y  $C_{cursado}$  es el tráfico total cursado por los usuarios conectados.

Esto equivale a decir que el nuevo usuario consumirá toda la capacidad libre de la celda, sin afectar significativamente al resto de los usuarios conectados. Además se está asumiendo que las tasas de conexión del nuevo usuario son similares a las de los usuarios ya conectados. Nótese que en la práctica esto puede que no se cumpla en todos los casos debido a que las tasas de conexión dependen de muchos factores tales como posiciones de los nodos, movimientos, interferencia o problemas de hardware que eventualmente puedan tener los equipos.

A continuación, se presenta un pequeño análisis que explica por qué es razonable suponer que la entrada del nuevo usuario no afecta a los usuarios que no están saturando, y por lo tanto únicamente afectaría a los usuarios que sí lo estuvieran.

Al ser el tráfico cursado por los nodos mayoritariamente TCP downlink, el hecho de que ingrese un nuevo usuario saturando a la red agrega un “cliente permanente”<sup>1</sup> al AP, único nodo que siempre está compitiendo por el canal. Esto genera una disminución en la frecuencia con que cada usuario recibe un paquete. Dicha disminución no tiene un impacto muy significativo en los usuarios no saturados ya que estos requieren paquetes del AP con mucho menor frecuencia que los nodos saturados. A estos últimos, sin embargo, debido a que siempre están esperando para recibir o enviar un paquete, este suceso les disminuiría notoriamente el tráfico total cursado.

En el caso 2 no se puede descartar a los usuarios inactivos tal como se hace en el caso 1 ya que estos son los únicos conectados y por ende quienes contienen la información disponible sobre el estado de la red. La estimación de throughput para un nuevo usuario será entonces igual a la del caso 1 excepto que el valor de capacidad total de la red ( $C_{total}$ ) es calculado en base a los usuarios inactivos.

Finalmente en el tercer caso se asume que el nuevo usuario será capaz de consumir la capacidad total que tendrá la red cuando este se conecte. Es decir, la estimación de throughput para el nuevo usuario será el valor de capacidad total

---

<sup>1</sup>Se dice que los nodos saturados son clientes permanentes debido a que están siempre esperando para enviar o recibir un paquete.

## 5.2. Obtención del modelo

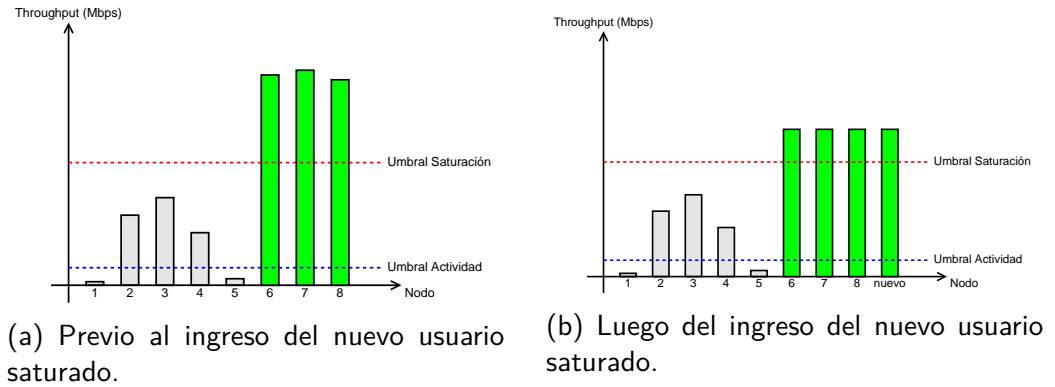


Figura 5.2: Reparto de throughput según modelo 2.i.

de la red ( $C_{total}$ ) obtenido mediante el modelo 1 y calculado únicamente en base a las tasas a las cuales se conecta. Para realizar la estimación se supone el mejor caso, por lo que se asumen velocidades de conexión máximas permitidas por la revisión del estándar 802.11 configurada para la red. En la tabla 5.1 se presentan los valores obtenidos.

Revisión	Basic Rate (Mbps)	Máx. Throughput (Mbps)
b	1	4.31
g	6	20.1
g	24	25.3

Tabla 5.1: Valores de throughput estimado para el nuevo usuario cuando no hay usuarios conectados a la red.

### 5.2.2. Escenario 2: Hay n usuarios saturados.

Para este escenario se proponen cuatro modelos que se presentan a continuación.

#### Modelo 2.i: Estimación a partir del throughput de los usuarios saturados

El primer modelo implementado para estimar la capacidad de throughput que obtendrá un nuevo usuario se basa en el throughput total cursado por los usuarios saturados. Consiste en asumir que el conjunto de nodos saturados se comportan como un solo nodo saturado, por lo cual al agregar un usuario se repartirá el tráfico de ellos entre los saturados de la nueva situación como se muestra en la figura 5.2.

Lo primero que se hace es identificar cuáles son los usuarios saturados. Haciendo el mismo análisis que fue presentado en la sección 5.2.1 se asume que el ingreso

## Capítulo 5. Segundo modelo de capacidad

de un nuevo usuario saturado afecta únicamente a los usuarios que ya estaban saturando. Puede verse además que debido a la equiprobabilidad de acceso al medio que otorga DCF todos los nodos saturados obtendrán aproximadamente el mismo throughput. Más aún, dado que se está trabajando bajo la hipótesis de tráfico TCP downlink, y por consiguiente el número de colisiones es bajo, el throughput total obtenido no se verá afectado por el ingreso del nuevo usuario. Entonces, una vez determinados los usuarios que saturan se calcula el tráfico total cursado por ellos, ( $C_{cursadoSat}$ ), y se lo reparte equitativamente entre los  $n + 1$  usuarios que habrá saturando luego del ingreso del nuevo.

El resultado de lo anterior es el primer modelo propuesto para estimar lo que obtendrá un nuevo usuario que se conecte a la red y se calcula de la siguiente forma:

$$C_{Modelo2i} = \frac{C_{cursadoSat}}{n + 1}$$

### Modelo 2.ii: Estimación a partir de las tasas de conexión de los usuarios saturados

El segundo modelo para la estimación del throughput que obtendrá el nuevo usuario surge a raíz del modelo anterior pero, en lugar de conservar el tráfico cursado por los usuarios saturados, propone conservar la *capacidad disponible* para ellos. Se denomina capacidad disponible para los usuarios saturados a la diferencia entre el valor de la capacidad total de la red (obtenido al aplicar modelo 1) considerando únicamente a los usuarios saturados y lo traficado por los usuarios no saturados. Una vez obtenida la capacidad disponible, se reparte entre la cantidad de usuarios saturados luego del ingreso de nuevo usuario. Se obtiene entonces:

$$C_{Modelo2ii} = \frac{C_{total-sat} - C_{cursadoNoSat}}{n + 1}, \quad (5.1)$$

donde  $C_{total-sat}$  es la capacidad total de la red calculada en base a los usuarios saturados,  $C_{cursadoNoSat}$  es el tráfico cursado por los usuarios no saturados y  $n$  es la cantidad de usuarios saturados.

Se calcula la capacidad total de la red teniendo en cuenta únicamente a los usuarios saturados ya que son estos los que se encuentran continuamente cursando tráfico. Esto en teoría implica que dichos usuarios son los que tienen mayor influencia en la capacidad total de la red.

A la estimación de la capacidad total de la red se le resta el throughput total que traficaron los usuarios no saturados ya que, al igual que para el caso en que no hay usuarios saturando, la llegada del nuevo usuario afecta principalmente a los usuarios saturados. Es decir, para obtener la capacidad disponible para los usuarios saturados, es necesario descontarle a la capacidad total de la red el tráfico ya consumido por los usuarios que no saturan.



## 5.2. Obtención del modelo

Finalmente, bajo la suposición de que el nuevo usuario se conecta a una tasa similar a la del resto de los usuarios saturados, se puede afirmar que se conserva la capacidad total de la red; es por esto que se reparte la capacidad disponible para los usuarios saturados entre la cantidad de usuarios saturados en la nueva situación.

### Modelo 2.iii: Estimación a partir de las tasas de conexión de todos los usuarios

Este modelo es equivalente al anterior con la única salvedad de que en este caso para el cálculo de la capacidad total de la red ( $C_{total}$ ) se consideran todos los usuarios en lugar de considerar únicamente a los usuarios saturados.

Se plantea como forma de verificar si efectivamente es representativo suponer que los usuarios saturados son los que más influencia tienen en cuanto a la capacidad de la red (hipótesis del modelo anterior).

### Modelo 2.iv: Estimación a partir del tráfico obtenido por encima del umbral

Este modelo surge de la idea de combinar la información que se obtiene a partir de las tasas de conexión y la información obtenida a partir de los throughputs de los usuarios saturados. Por un lado utiliza la estimación de capacidad del modelo 1 para tener una idea de cuánto sacaría cada usuario si estuviesen todos (incluyendo el nuevo) saturando, y luego ajusta este valor en función del tráfico que efectivamente se sabe que están sacando los saturados.

Para el desarrollo del modelo se definen dos parámetros,  $Umbral_{todos}$  y  $Umbral_{todos+1}$ , que representan cuánto sacaría cada usuario si estuvieran todos saturando en la situación previa y posterior a la entrada del nuevo usuario respectivamente.

$$Umbral_{todos} = \frac{C_{total}}{u} \quad (5.2)$$

$$Umbral_{todos+1} = \frac{C_{total}}{u+1} \quad (5.3)$$

Aquí  $u$  es la cantidad de usuarios conectados al inicio y  $C_{total}$  es la capacidad total de la red calculada en base a todos los usuarios.

Se asume entonces que los usuarios saturados obtendrán como mínimo el throughput que obtendrían todos los usuarios (incluyendo al nuevo) si estuvieran saturando ( $Umbral_{todos+1}$ ). Se asume también que se mantendrá constante lo que trafican los saturados por encima de  $Umbral_{todos}$ , y que este tráfico cursado además, una vez ingresado el nuevo usuario, se reparte en partes iguales entre los saturados como se representa en la figura 5.3.

## Capítulo 5. Segundo modelo de capacidad

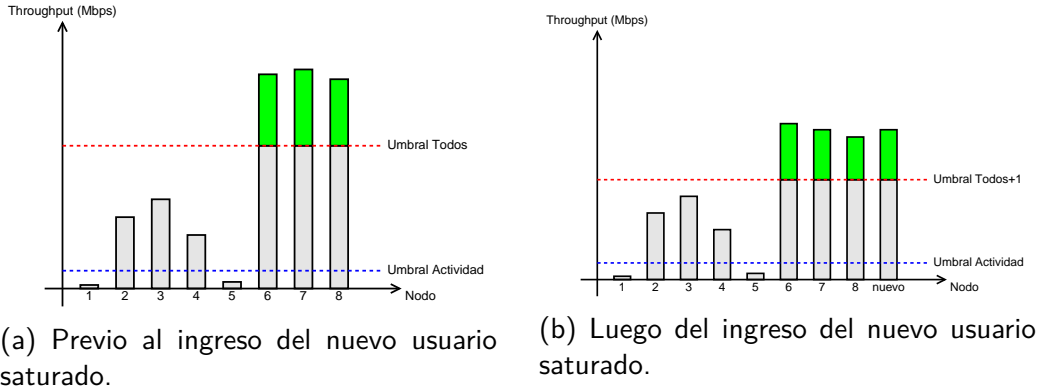


Figura 5.3: Reparto de throughput según modelo 2.iv.

Una vez obtenido  $Umbral_{todos+1}$  se quiere determinar cuánto tráfico por encima de él es posible obtener, para lo cual se calcula cuánto se obtuvo por encima de  $Umbral_{todos}$  previo al ingreso del nuevo usuario y se asume que este valor no se ve afectado por su ingreso a la red. Así, el tráfico total por encima del límite de saturación ( $C_{sobreSat}$ ) se calcula en base al tráfico cursado por los usuarios saturados de la siguiente manera:

$$C_{sobreSat} = \sum_{i=1}^n (C_i - Umbral_{todos})$$

donde:

- $n$  es la cantidad de usuarios saturados.
- $C_i$  es el tráfico cursado por cada usuario saturado.

Finalmente, se debe repartir este tráfico entre los usuarios saturados luego del ingreso del nuevo, y añadirse a  $Umbral_{todos+1}$ , obteniendo así la estimación del throughput para el nuevo usuario:

$$C_{Modelo2iv} = Umbral_{todos+1} + \frac{C_{sobreSat}}{n + 1}$$

Nótese que para el cálculo de  $Umbral_{todos+1}$  se tiene en cuenta al nuevo usuario ya que se pretende hallar el punto de equilibrio de throughput de la red con su presencia, es decir el throughput que obtendrían todos los usuarios (incluyendo el nuevo) si estuvieran todos saturando. Para tener en cuenta al nuevo usuario es necesario suponer un valor para la tasa a la que se conecta. Recordando que la implementación de este proyecto está orientada a redes del Plan Ceibal tiene sentido suponer que el nuevo usuario se comporta de manera similar a los usuarios ya conectados. Esto se debe a que todos los usuarios utilizan equipos con prestaciones

## 5.2. Obtención del modelo

idénticas o muy similares y que todos tienen la misma potencia de señal recibida. Dicha suposición equivale a calcular la capacidad total de la red teniendo en cuenta sólo a los usuarios conectados y para incluir al nuevo usuario se reparte dicha capacidad entre todos los usuarios conectados tras la entrada del nuevo usuario.

A continuación se analiza el throughput total obtenido por los usuarios saturados antes y después del ingreso del nuevo según este modelo.

Previo a la entrada del nuevo usuario se tiene:

$$Thr_{satAnt} = SumaArriba + n \cdot Umbral_{todos} \quad (5.4)$$

donde  $SumaArriba$  es la porción de tráfico de los usuarios saturados por encima de  $Umbral_{todos}$ , y  $n$  es la cantidad de usuarios saturados. Análogamente, luego de la entrada del nuevo usuario se tiene:

$$Thr_{satDesp} = SumaArriba + (n + 1) \cdot Umbral_{todos+1} \quad (5.5)$$

La figura 5.4 presenta un ejemplo del tráfico en la red antes y después del ingreso del nuevo usuario según lo esperado por este modelo.

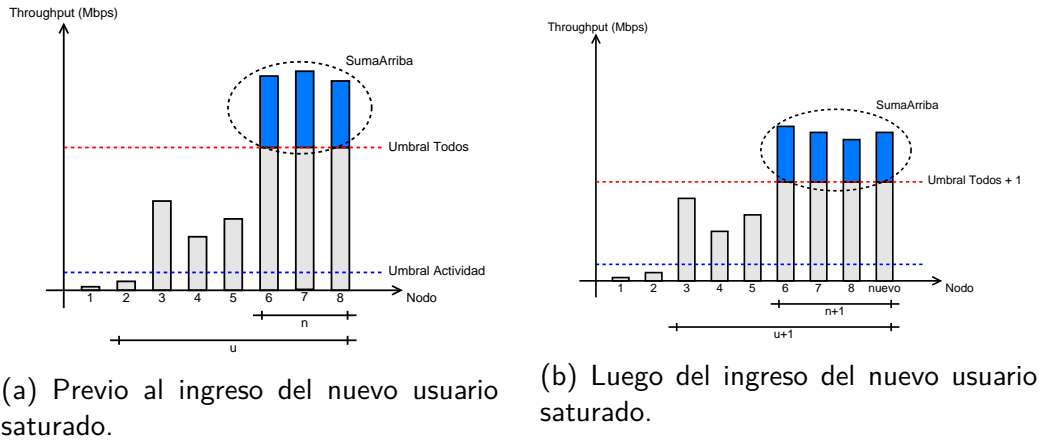


Figura 5.4: Diagrama de tráfico por usuario, antes y después del ingreso del nuevo usuario saturado.

De las ecuaciones 5.2 y 5.3 se puede despejar:

$$Umbral_{todos+1} = \frac{u}{u + 1} \cdot Umbral_{todos}$$

y sustituyendo en la ecuación 5.5 se obtiene:

$$Thr_{satDesp} = SumaArriba + \frac{(n + 1) \cdot u}{u + 1} \cdot Umbral_{todos}$$

## Capítulo 5. Segundo modelo de capacidad

Para poder analizar la relación entre  $Thr_{satDesp}$  y  $Thr_{satAnt}$ , teniendo en cuenta que ambas poseen en común el término  $SumaArriba$ , solo resta conocer la relación entre los otros sumandos. Para esto calculamos el cociente (K) entre dichos sumandos, obteniendo:

$$K = \frac{n+1}{n} \cdot \frac{u}{u+1}$$

que se puede escribir como:

$$K = \frac{1 + \frac{1}{n}}{1 + \frac{1}{u}} \quad (5.6)$$

A partir de la ecuación 5.6 se puede ver que al ser  $u \geq n$  se cumplirá que:

$$1 \leq K < 2$$

valiendo 1 para el caso en que  $u = n$  y tendiendo a 2 cuando  $n = 1$  y  $u \gg 1$ .

Se puede deducir también que cuanto menor sea  $n$  y/o mayor sea  $u$ , mayor será  $K$ . En otras palabras el nuevo throughput total obtenido por los usuarios saturados ( $Thr_{satDesp}$ ) será siempre mayor o igual que el obtenido antes del ingreso del nuevo usuario ( $Thr_{satAnt}$ ). Esto permite concluir que el nuevo usuario es capaz de consumir parte de la capacidad ociosa que tiene la celda debido a la presencia de usuarios no saturados.

Para cuantificar esta diferencia se restan las ecuaciones 5.5 y 5.4 obteniéndose:

$$Diff_{Thr} = Thr_{satDesp} - Thr_{satAnt} = \left( \frac{n+1}{u+1} - \frac{n}{u} \right) \cdot C_{total} \quad (5.7)$$

Se puede ver que para  $u = n$  se obtiene  $Thr_{satDesp} = Thr_{satAnt}$ . Por otro lado, fijando  $n = 1$ , los máximos de esta diferencia obtenidos como:

$$\left. \frac{\partial(Thr_{satDesp} - Thr_{satAnt})}{\partial u} \right|_{n=1} = \left( \frac{-(n+1)}{(u+1)^2} + \frac{n}{u^2} \right) \Big|_{n=1} \cdot C_{total} = 0$$

se dan para  $u_{max} = 1 \pm \sqrt{2}$ . Dado que  $u$  es un número entero y que la diferencia es máxima para valores mínimos de  $n$ , se evalúa 5.7 en  $u = 2$  y  $u = 3$  para  $n = 1$  por ser los enteros más cercanos a  $u_{max}$ , obteniéndose para estos:

$$Diff_{Thr}(n = 1, u = 2) = Diff_{Thr}(n = 1, u = 3) = \frac{C_{total}}{6}$$

Por esto se puede concluir que al agregar usuarios saturados aumenta el throughput total que estos obtienen como máximo una sexta parte de la capacidad total de red.

Este análisis contradice la premisa fundamental del modelo 2.i que se basaba en que el throughput obtenido por los usuarios saturados permanecía constante

### 5.3. Implementación del modelo

luego del ingreso del nuevo usuario.

Cabe aclarar que el aumento de throughput obtenido por los usuarios saturados disminuye fuertemente con el aumento de la cantidad de usuarios saturados y no saturados. Este comportamiento se ve reflejado en la figura 5.5 donde se grafica la variación del aumento de throughput con el aumento de la cantidad de usuarios ( $u$ ) para distintas cantidades de usuarios saturados ( $n$ ).

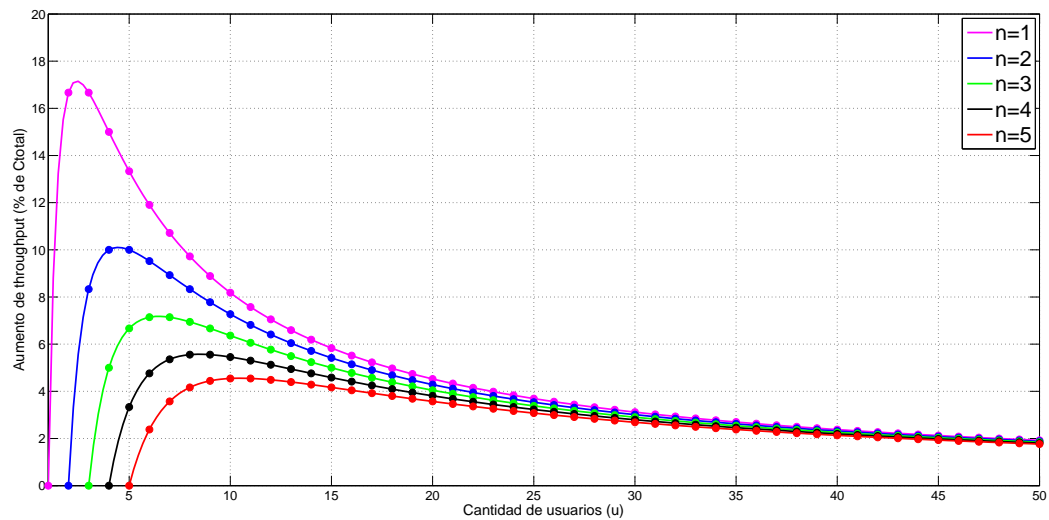


Figura 5.5: Aumento de throughput de los usuarios saturados al ingresar uno nuevo.

### 5.3. Implementación del modelo

Para la validación de este nuevo modelo de capacidad es necesaria la implementación de varios scripts en Shell que se ejecutan tanto del lado de los clientes como del lado del servidor.

El primero de estos scripts, *genero\_trafico.sh* es ejecutado en cada uno de los clientes conectados a la red y su función es la de generar un tráfico aleatorio en la red. Luego, mediante la ejecución en el servidor de *modelo2-consultaSNMP\_rate.sh* se obtienen los datos de tasas de conexión y tráfico cursado por cada uno de los nodos. Esta información es luego utilizada por el script *procesamiento.sh* que determina todos los valores de umbrales, capacidades y throughputs obtenidos que permiten el cálculo de la estimación para cada uno de los modelos propuestos. A su vez, devuelve la estimación obtenida mediante el modelo 2.ii. Se implementa también un script *main.sh* que mediante la invocación del script de consultas SNMP y el de procesamiento de datos, realiza el ciclo completo dando como resultado la estimación de capacidad para un nuevo usuario.

De manera adicional, se implementa un script *analizo\_rates.sh* cuya función es

## Capítulo 5. Segundo modelo de capacidad

la de obtener la variación de las tasas de conexión de un nodo particular en una prueba específica. Su función principal es la detección rápida de cambios en los valores de las tasas de conexión que pueden perjudicar el throughput total obtenido y facilitar la etapa de análisis de datos permitiendo realizar posibles ajustes a los modelos.

El código de cada uno de estos scripts puede ser consultado en el CD adjunto a esta documentación.

### `genero_trafico.sh`

Dado que el estado de la red para el escenario del modelo 2 es desconocido y variable, para la validación del modelo en el laboratorio se implementa un script que genera en la red un tráfico aleatorio y en principio desconocido.

Para esto se añaden al repositorio de archivos utilizado para las pruebas de laboratorio del modelo 1, una gran variedad de archivos de distintos tamaños. Luego, ejecutando el script `genero_trafico.sh` en las terminales de los clientes conectados, se seleccionan algunos de esos archivos, y se los descarga sucesivas veces dejando entre descarga y descarga un tiempo de espera ingresado por el usuario al momento de la ejecución. De esta manera se genera un tráfico que intenta simular el comportamiento de los nodos no saturados, es decir, aquellos cuyo comportamiento es, en principio, desconocido.

### `modelo2-consultaSNMP_rate.sh`

El funcionamiento de este script es idéntico al del bloque de consultas SNMP del script `consultaSNMP_rate.sh` implementado para la validación del modelo 1 y detallado en la sección 4.3. Tiene como entradas la duración total de la prueba, el período de interconsulta, y la versión del estándar utilizada en la red. Sus salidas son nuevamente los bytes que llevaba cursado cada uno de los nodos al principio y al final de las pruebas y sus variaciones de tasas de transmisión y recepción por intervalo de tiempo como se muestra en el diagrama de la figura 5.6. Estas salidas, además de ser desplegadas en pantalla, son respaldadas en un directorio de salidas históricas para un posible futuro reprocesamiento en caso de ser necesario.

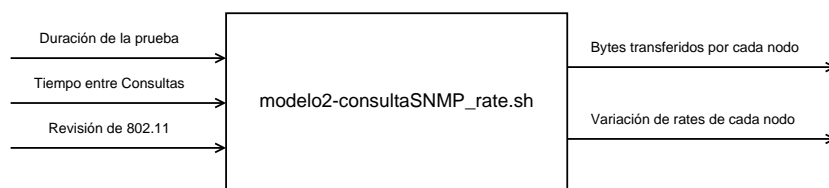


Figura 5.6: Diagrama del script `modelo2-consultaSNMP_rate.sh`

## procesamiento.sh

Una vez obtenidos los datos de variación de tasas de conexión y bytes cursados por cada uno de los nodos conectados a la red, se tiene la información suficiente como para realizar cualquiera de las estimaciones propuestas en la sección anterior.

Este script recibe como entradas la duración total de la prueba, el período de interconsulta, la versión del estándar utilizada y el valor del umbral de actividad, a partir de las cuales procesa la información obtenida por *modelo2-consultaSNMP\_rate.sh* y determina la estimación según el modelo 2.ii. A su vez, calcula y despliega todos los indicadores de umbrales, throughputs medidos y esperados según el modelo 1, necesarios para el reprocesamiento externo de las medidas y el cálculo de la estimación según los otros modelos propuestos.

En primera instancia, el cálculo del throughput cursado por cada usuario se realiza en base a la diferencia entre los bytes cursados al final de la prueba y los bytes cursados al inicio, y se asume que se mantuvo constante durante todo el transcurso de la prueba. Dado que en este nuevo escenario el comportamiento de los usuarios puede ser cualquiera, se asume que estos podrían llegar a desconectarse temporalmente de la red o conectarse en el transcurso de la prueba, por lo cual debe hacerse un tratamiento especial que contemple dichos casos. Se pueden distinguir entonces los siguientes comportamientos a los que se trata de diferentes maneras a la hora del cálculo del throughput cursado:

- **Caso 1:** El usuario estuvo presente durante todas las consultas SNMP (Fig. 5.7a).
- **Caso 2:** El usuario estuvo presente en la primera y última consulta SNMP, pero perdió alguna consulta intermedia (Fig. 5.7b).
- **Caso 3:** El usuario no estuvo presente durante la primera consulta, pero sí estuvo durante la última (Fig. 5.7c).
- **Caso 4:** El usuario no estuvo en la última consulta (Fig. 5.7d).

En principio, el caso 1 es el comportamiento deseado y el throughput total medido para ese usuario se calculará como:

$$\text{Throughput} = \frac{\text{Bytes}(t_f) - \text{Bytes}(t_0)}{t_f - t_0}$$

donde  $t_0$  es el tiempo de inicio de la prueba (en segundos),  $t_f$  es el tiempo final de la prueba (en segundos),  $\text{Bytes}(t_0)$  son los bytes que ha descargado el nodo hasta el inicio de la prueba,  $\text{Bytes}(t_f)$  son los bytes que ha descargado el nodo hasta el final de la prueba. Debido a que el AP comienza desde cero una nueva cuenta de bytes cursados cada vez que el nodo se conecta a la red, eventualmente puede suceder que la diferencia  $\text{Bytes}(t_f) - \text{Bytes}(t_0)$  dé como resultado un entero negativo. Esto puede suceder si el nodo se desconecta de la

Capítulo 5. Segundo modelo de capacidad

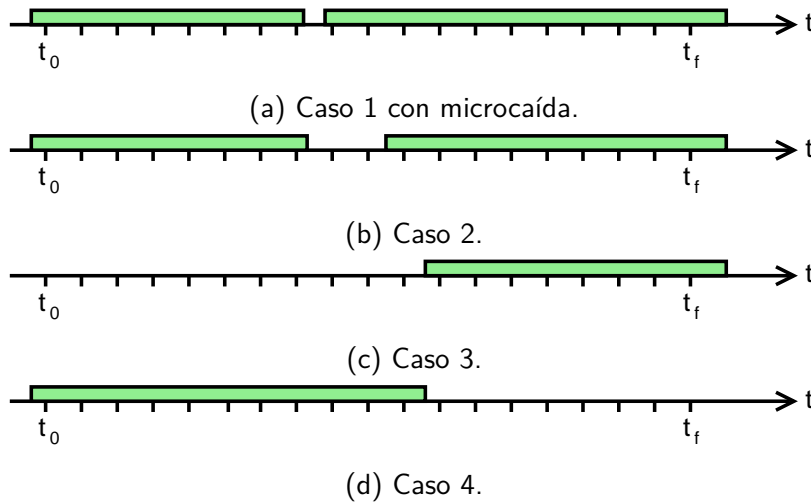


Figura 5.7: Distintos casos de desconexiones temporales. En verde se indican los intervalos de tiempo en los que el nodo estaba conectado.

red luego de una consulta pero vuelve a conectarse antes de la siguiente<sup>2</sup>. En este caso el throughput cursado se aproxima como:

$$Throughput = \frac{Bytes(t_f)}{t_f - t_0}$$

Puede suceder también que, a pesar de haber una micro-caída, si el valor de  $Bytes(t_0)$  obtenido es menor a los bytes cursados desde la reconexión hasta  $t_f$ , la diferencia  $Bytes(t_f) - Bytes(t_0)$  tome un valor positivo lo cual provoca que la micro-caída sea indetectable. Esto podría llegar a suceder si la micro-caída se genera poco tiempo después de una conexión.

Para el caso 2, donde la desconexión del nodo fue detectada por su ausencia durante al menos una consulta, el throughput total cursado por él debe tener en cuenta lo traficado antes y después de la desconexión. En la etapa de obtención de datos se consulta únicamente por los bytes transmitidos al principio y al final de la prueba de forma de disminuir la cantidad de consultas y con esto el tiempo de ejecución del script `consultaSNMP_rate.sh`. Esto hace que no se tenga información sobre lo traficado antes de la desconexión, por lo que solo se toma en cuenta lo traficado luego de la última conexión. Esta aproximación tendrá un impacto significativo cuando quien sufra la desconexión temporal sea un usuario saturado. De esta manera se obtiene la siguiente aproximación para el throughput traficado:

$$Throughput = \frac{Bytes(t_f)}{t_f - t_0}$$

<sup>2</sup>Se dirá que en este caso el nodo sufrió una micro-caída que fue imposible detectar mediante las consultas SNMP pero fue detectada a la hora del cálculo del throughput.



### 5.3. Implementación del modelo

En el caso 3, debido a la falta de información acerca de  $Bytes(t_0)$ , la única expresión posible para el throughput cursado es nuevamente:

$$Throughput = \frac{Bytes(t_f)}{t_f - t_0}$$

Finalmente, si el nodo no estuvo presente al momento de la última consulta es imposible según esta implementación obtener una aproximación para el throughput obtenido por lo cual se descarta a dicho nodo de cualquier tipo de cálculo.

Como puede verse este tratamiento puede introducir algunos errores en el cálculo del throughput que dependen de la cantidad de desconexiones que sufran los usuarios en la red y el momento en el cual ocurren. Disminuir el tiempo de interconsulta no sólo mejora la detección de cambios de tasas de transmisión y recepción, sino que también puede hacer que algunas de las desconexiones temporales de usuarios sean detectadas y así tener una mejor estimación del tráfico cursado. Sin embargo, hacer eso también implica que el servidor deba hacer una mayor cantidad de consultas SNMP con lo cual aumenta la cantidad de datos a procesar y por consiguiente aumenta el tiempo de ejecución del script. Es por esto que se debe elegir un tiempo de interconsulta que tome en cuenta ambos factores.

Luego de calculados los throughputs medidos para cada nodo el script determina cuales están por debajo del umbral de actividad ingresado y los declara como usuarios inactivos. Asimismo, en base a las tasas de conexión de todos los usuarios determina el umbral de saturación con lo cual define qué nodos están saturando y qué nodos no. Finalmente, en base a los throughputs de los nodos no saturados y las tasas de conexión de los saturados, estima el máximo throughput posible para el nuevo usuario según la ecuación 5.1.

En la figura 5.8 se muestra un ejemplo de la salida del script *procesamiento.sh* para una prueba realizada en el laboratorio.

#### analizo\_rates.sh

Para la etapa del análisis de los datos obtenidos y posibles ajustes a los modelos, es necesario contar con un script que indique rápidamente cómo fue la variación de las tasas de conexión para un nodo específico durante el transcurso de una prueba. Así, las entradas de este script son la fecha y hora de la prueba, y el número de usuario cuyas tasas quieran desplegarse. En la figura 5.9 se muestra un ejemplo de la salida del script para una prueba de laboratorio.

## Capítulo 5. Segundo modelo de capacidad

```
===== 02-09-15----19-18 =====
Tiempo de ejecucion: 185.305636695 segundos

User 1 MAC: 00 21 85 F5 7A 1E --- Rate: 11386.22 Kbps
User 2 MAC: 28 E3 47 4F 6E EF --- Rate: 695.17 Kbps
User 3 MAC: 2C D0 5A D1 E5 09 --- Rate: 0 Kbps
User 4 MAC: 74 F0 6D 44 81 54 --- Rate: 3595.83 Kbps
User 5 MAC: 78 92 9C A4 76 3C --- Rate: .08 Kbps
User 6 MAC: 78 92 9C A8 63 84 --- Rate: 801.14 Kbps

Throughput Total Medido: 16478.44 Kbps
Throughput Total Esperado: 18180.19 Kbps

SiMonDiC que un nuevo usuario que se conecte a la red deberia obtener aproximadamente: 6543.98 Kbps

No se conecto ningun usuario nuevo durante la prueba.

Los siguientes 1 usuarios sufrieron desconexiones temporales durante el transcurso de la prueba:
78 92 9C A4 76 3C

Los siguientes 1 usuarios tienen un throughput por encima del umbral (6543.98 Kbps):
00 21 85 F5 7A 1E

Umbral Todos + 1: 3618.74 Kbps
Umbral Saturados: 13087.97 Kbps
Umbral Saturados + 1: 6543.98 Kbps

Esperado Todos: 18093.71 Kbps
Esperado Saturados: 13087.97 Kbps
```

Figura 5.8: Ejemplo de salida del script *procesamiento.sh*

Down Rate	Up Rate
48	6
48	12
48	12
48	12
48	12
48	12
48	12
48	18
48	18
48	18
48	18
48	18
48	6
48	18
48	18
48	18
48	18

Figura 5.9: Ejemplo de salida del script *analizo\_rates.sh*

### 5.4. Pruebas en laboratorio y validación de modelos

Para realizar la validación de los modelos en un entorno controlado, se genera una red con las mismas características de tráfico que se espera encontrar en la red de una escuela. Para ello se contemplan escenarios en donde habrá mezcla de usuarios no saturados, usuarios saturados y usuarios que no cursan ningún tipo de tráfico.

Para simular usuarios no saturados, se ejecuta en algunos clientes el script *genero\_trafico.sh* que los pone a descargar “para siempre” archivos de diferentes tamaños con determinados tiempos de espera entre una descarga y otra. De esta manera se genera un tráfico distinto para cada uno de ellos, generando en conjunto

## 5.4. Pruebas en laboratorio y validación de modelos

un tráfico significativo similar al que se espera encontrar en una red estándar. Los clientes saturados se simulan poniendo los nodos que harán de saturados a descargar un archivo de tamaño aproximado 1 GB, que, a los efectos de la duración de la prueba, puede ser considerado como un archivo infinitamente grande. Finalmente, para simular los clientes inactivos, simplemente se conectan algunos nodos a la red sin ponerlos a descargar nada.

Para evaluar la performance de los modelos las pruebas se hacen sobre escenarios con distinta cantidad de nodos saturados. Es decir, se evalúa cómo se comportan los modelos cuando entra un nuevo usuario a saturar a una red en la que al inicio hay determinada cantidad de usuarios no saturados, cierta cantidad de usuarios conectados sin cursar tráfico y  $n$  (entero entre 1 y 3) usuarios saturados. Las pruebas de laboratorio se realizan para escenarios con un máximo de 3 usuarios saturados al inicio.

Para cada valor de  $n$  se realizan 10 pruebas, que consisten en aplicar a la red descrita el script *modelo2-consultaSNMP\_rate.sh* mediante el cual se obtiene la información de tráfico cursado y tasas de conexión para cada nodo. A partir de esta información, el script *procesamiento.sh* obtiene el valor de estimación de throughput para el nuevo usuario según el modelo 2.ii y se obtienen además los datos necesarios (umbrales de saturación, throughput medido, throughput esperado según el modelo 1) para realizar los cálculos de la estimación de throughput  $C_{Modelo2}$  según el resto de los modelos.

Una vez obtenida la estimación que devuelve el script *procesamiento.sh* se conecta el nuevo usuario a la red, se lo pone a descargar el mismo archivo que descargaban los nodos saturados y se mide cuál es el throughput que efectivamente obtiene. Esta medición se realiza considerando el tráfico cursado por este nuevo nodo en un intervalo de 3 minutos. Comparando este valor medido con los valores estimados por cada modelo se obtiene el error relativo de cada estimación. Es deseable que las curvas de los errores relativos sean lo más planas y cercanas a cero posibles.

Para la validación de este nuevo modelo se debe elegir un tiempo de duración de prueba más prolongado que en las pruebas del modelo 1 para que el efecto de la irregularidad del tráfico cursado por los usuarios no saturados introduzca los menores errores posibles. A su vez, elegir un tiempo excesivamente grande no solo hace que la implementación deje de ser práctica, sino que también puede introducir otros errores que van asociados a las desconexiones temporales de los usuarios de la red. Por esto se elige un tiempo de duración de prueba de 3 minutos.

En las siguientes secciones se realiza la validación de los modelos propuestos distinguiendo nuevamente entre el escenario en que hay  $n$  usuarios saturados y el escenario en que no hay ninguno. Se pretende determinar cuál es el modelo con mejor desempeño para luego verificar si es efectivamente el que mejor se desem-

## Capítulo 5. Segundo modelo de capacidad

peña en las pruebas de campo.

### 5.4.1. Escenario sin usuarios saturados

En la sección 5.2.1 se propuso un único modelo para estimar cuál es el máximo throughput que puede obtener un nuevo usuario que ingrese a saturar en una red en la que inicialmente solo hay usuarios inactivos o no saturados.

En la figura 5.10 se muestran los resultados obtenidos tras la comparación de la capacidad estimada para el nuevo usuario con el tráfico que efectivamente obtuvo tras conectarse a la red, donde el error relativo entre lo estimado y lo medido se calcula de la siguiente manera:

$$ErrRel = \frac{C_{Modelo2} - C_{medido}}{C_{medido}} \quad (5.8)$$

Se puede ver que las estimaciones realizadas estuvieron en todas las pruebas muy cercanas al throughput efectivamente obtenido por el nuevo usuario, con errores relativos en casi todos los casos menores al 10%. Sin embargo, en casi toda las pruebas el throughput obtenido estuvo por debajo de la estimación. En la sección 5.6 se analiza con mayor profundidad la razón de esto.

### 5.4.2. Escenario en que hay $n$ usuarios saturados

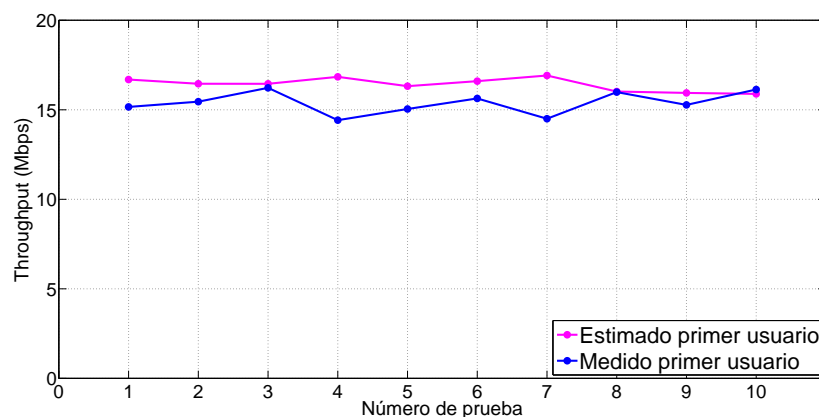
En la sección 5.2.2 se propusieron cuatro modelos para la estimación del throughput que obtiene un nuevo usuario cuando ingresa a una red en la cual ya hay  $n$  usuarios saturados.

En las figuras 5.11a, 5.11b y 5.11c se presentan los errores relativos obtenidos por cada uno de los cuatro modelos propuestos para el caso en que inicialmente hay uno, dos y tres usuarios saturando respectivamente. Es fácil ver que no existe un modelo que tenga mejor desempeño que los tres restantes en todos los casos. Esto implica que se deben analizar más en detalle los resultados obtenidos para poder discernir qué modelo es el más adecuado.

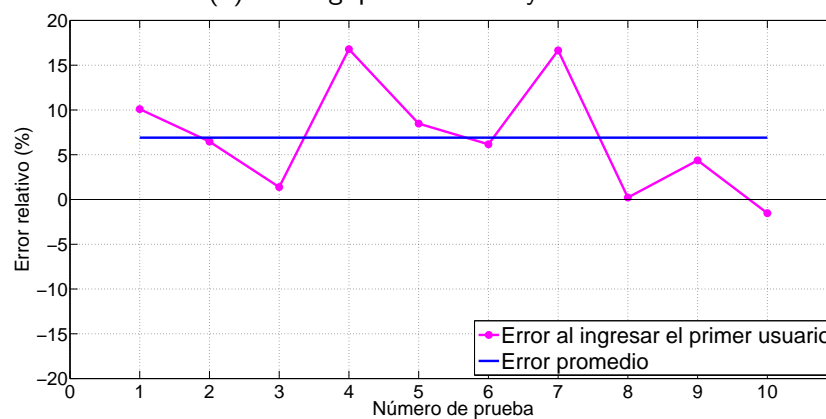
La figura 5.11a permite observar que el modelo 2.i obtiene errores relativos mucho mayores que el resto de los modelos para cada una de las 10 pruebas. Este es el único caso en que un modelo tiene un desempeño mucho peor que los otros tres en todas las pruebas, obteniendo errores relativos entre 8% y 22% mayores. A su vez, en las figuras 5.11b y 5.11c se observa que este modelo nunca es el que presenta menor error relativo.

El hecho de que el modelo 2.i haya subestimado el throughput en el 90% de las pruebas resulta consistente con el análisis hecho en 5.2.2, donde se presume que el throughput obtenido por los usuarios saturados incrementa con el ingreso

## 5.4. Pruebas en laboratorio y validación de modelos



(a) Throughput estimado y medido.



(b) Error relativo.

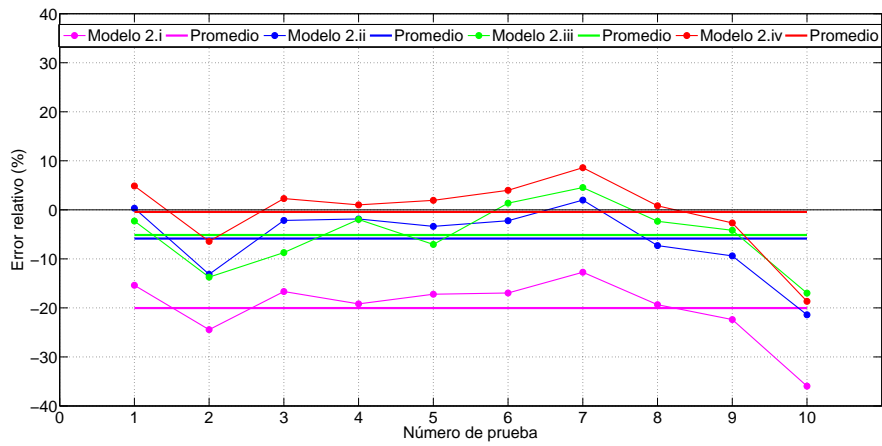
Figura 5.10: Resultados obtenidos aplicando el modelo 2 en el laboratorio para el caso en que no hay usuarios saturados.

del nuevo usuario. La diferencia entre las estimaciones de estos modelos se hace más evidente en el caso en que hay un único usuario saturado al inicio (ver figura 5.11a), lo que condice nuevamente con el análisis según el cual el mayor incremento de throughput se da con la menor cantidad de usuarios saturados (ver figura 5.5).

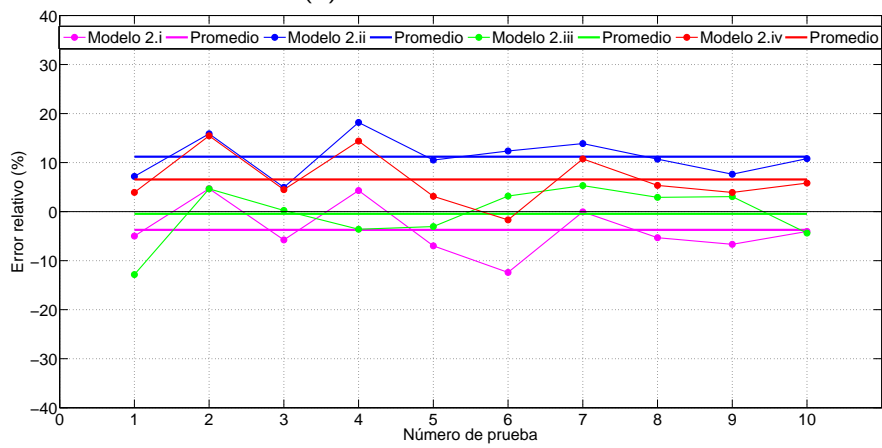
En base a las observaciones anteriores y considerando cuánto mejor es el desempeño del modelo 2.iv respecto del 2.i, resulta razonable, en principio, descartar el modelo 2.i.

Los modelos 2.ii y 2.iii estiman el tráfico del nuevo usuario a partir de la capacidad total de la red, el tráfico de los usuarios no saturados y la cantidad de usuarios saturados. La única diferencia entre estos es que el modelo 2.ii realiza la estimación en base a los usuarios saturados mientras que el 2.iii lo hace en base al total de los usuarios. En la figura 5.11a se puede ver que ambos modelos presentan un desempeño muy similar, mientras que las figuras 5.11b y 5.11c muestran que

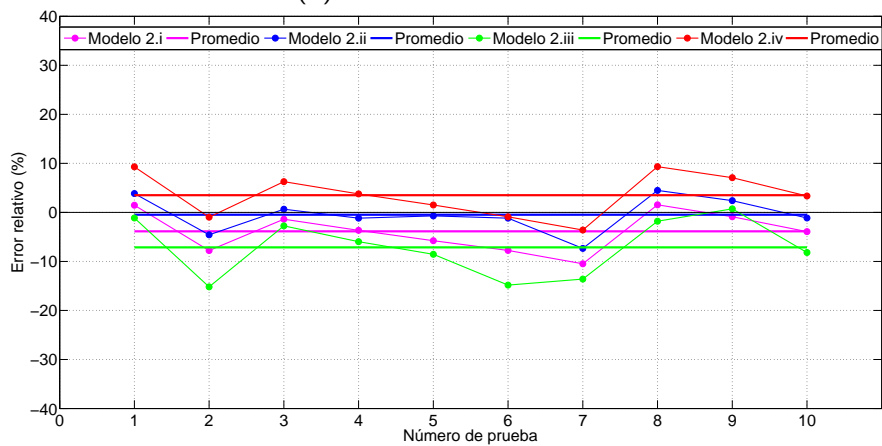
## Capítulo 5. Segundo modelo de capacidad



(a) 1 saturado al inicio.



(b) 2 saturados al inicio.



(c) 3 saturados al inicio.

Figura 5.11: Comparación de los errores relativos obtenidos por los modelos propuestos.

#### 5.4. Pruebas en laboratorio y validación de modelos

el modelo con mejor desempeño es el modelo 2.iii en el primer caso, y el modelo 2.ii en el segundo, en ambos casos con errores relativos promedios muy cercanos a cero. Estas similitudes no permiten en principio descartar a ninguno de ellos. Por esto, a continuación se comparan ambos modelos con el modelo 2.iv.

Dado que los modelos 2.ii, 2.iii y 2.iv presentan errores relativos muy similares resulta complicado compararlos gráficamente. Es por esto que en la tabla 5.2 se presenta un resumen de los errores relativos promedios y las desviaciones estándar de los modelos para los tres casos (uno, dos y tres usuarios saturados al inicio). Se puede ver que cada uno de ellos presenta el mejor desempeño para un caso distinto obteniendo un error relativo prácticamente nulo. El modelo 2.iv tiene la ventaja de presentar el menor valor de pico de error relativo promedio (6.55%). En cuanto a la desviación estándar no se perciben diferencias significativas.

n	modelo 2.ii		modelo 2.iii		modelo 2.iv	
	E R (%)	D E (%)	E R (%)	D E (%)	E R (%)	D E (%)
1	-5.85	7.14	-5.13	6.61	-0.43	7.59
2	11.21	4.05	-0.45	5.6	6.55	5.36
3	-0.48	3.63	-7.13	5.92	3.51	4.5

Tabla 5.2: Errores relativos y desviación estándar de los modelos 2.ii, 2.iii y 2.iv con n la cantidad de usuarios saturados al inicio de la prueba

Otro factor a tener en cuenta a la hora de inclinarse por un modelo, es si este representa un mejor o un peor caso del valor que se quiere estimar. En este caso, al tratarse de valores de throughput, se entiende por mejor caso a una estimación de una cota superior del throughput que es posible obtener. En la tabla 5.3 se presenta para cada modelo qué porcentaje de las pruebas fue estimado por encima de lo que efectivamente obtuvo el nuevo usuario. Se puede observar que el único modelo que presenta una clara tendencia en este sentido es el 2.iv, representando entonces un mejor caso del throughput a estimar.

n	modelo 2.ii (%)	modelo 2.iii (%)	modelo 2.iv (%)
1	20	20	70
2	100	60	90
3	40	10	70

Tabla 5.3: Porcentaje de estimaciones por encima del valor obtenido

Con lo visto hasta ahora, se puede concluir que cualquiera de los modelos 2.ii, 2.iii y 2.iv son modelos válidos para estimar el máximo throughput posible para un nuevo usuario de la red. En particular, teniendo en cuenta que el modelo 2.iv tiene

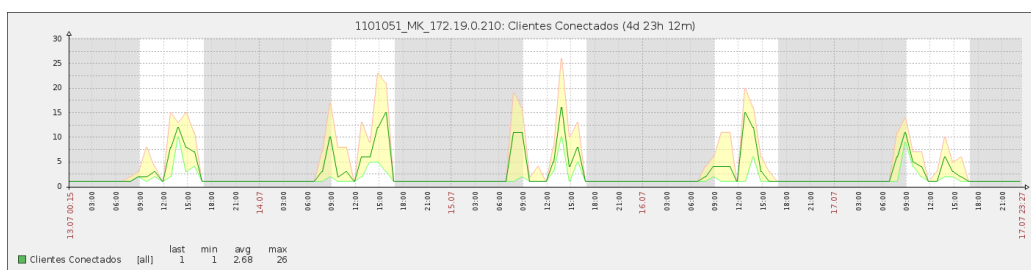
## Capítulo 5. Segundo modelo de capacidad

un comportamiento más consistente a lo largo de los tres casos, el menor valor de pico en el error relativo y representa un mejor caso de la estimación deseada, es razonable elegirlo como el modelo más adecuado para la estimación.

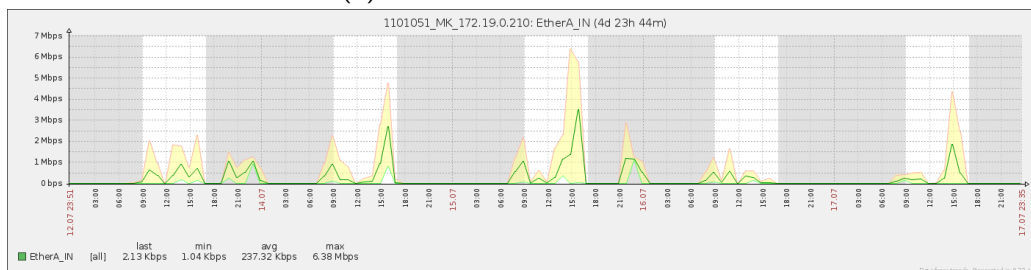
### 5.5. Pruebas en escuela

Tras haber evaluado el desempeño de los distintos modelos en un entorno de laboratorio, donde gran cantidad de factores están controlados, es necesario evaluar el desempeño de los modelos en pruebas de campo. Tras establecer contacto con personal de Plan Ceibal y con la información de que en la Escuela N° 51 República del Paraguay, ubicada en Julio César esq. Avda. Gral. Rivera, se utilizan APs MikroTik, se decide ir allí a hacer las pruebas de campo del modelo 2.

Para determinar el mejor momento del día para realizar las pruebas se analizan las figuras 5.12a y 5.12b que muestran la evolución temporal de la cantidad de usuarios conectados y throughput de bajada para el AP de la escuela con mayor tráfico. Aquí se grafican datos promediados cada 5 minutos que luego se procesan para el almacenamiento de su histórico. Se ven tres curvas que corresponden al valor máximo, mínimo y promedio por hora. En estas se puede apreciar que todos los días se produce un pico de usuarios y tráfico en el horario comprendido entre las 14:00 y las 16:00, por lo que se concluye que ese es el mejor horario para la realización de las pruebas.



(a) Clientes conectados al AP.



(b) Tráfico de downlink del AP.

Figura 5.12: Variación de clientes conectados y tráfico de downlink para el AP.



## 5.5. Pruebas en escuela

Para la realización de las pruebas se asignó una IP perteneciente a la red local de la escuela al servidor Debian sobre el cual se corren los scripts para que de esta manera sea posible acceder al Access Point y realizar las mediciones de manera análoga a como se hizo en las pruebas de laboratorio.

La estimación para la capacidad del nuevo usuario se realiza nuevamente en base al procesamiento de las tasas de conexión y throughputs medidos de los clientes durante intervalos de 3 minutos. Inmediatamente luego de la estimación se fuerza el ingreso de un nuevo usuario y se lo pone a descargar un archivo del servidor local de imágenes de software. Aquí se alojan actualizaciones de los sistemas operativos de los distintos modelos de equipos utilizados por los estudiantes, por lo cual la gran mayoría de los archivos son de tamaños superiores a los 500 MB. Para estas pruebas se escogió un archivo de aproximadamente 900 MB de tamaño.

En las primeras pruebas realizadas se trabajó sobre la red en su estado usual, es decir sin organizar ningún tipo de actividad específica con las maestras o alumnos durante las mediciones. A pesar de que estas pruebas fueron realizadas en el horario en el que se encontraba el pico máximo de tráfico, se notó que dicho pico era muy corto y a veces no tan alto como se esperaba, lo cual hacía difícil obtener una buena cantidad de pruebas con las cuales validar los modelos. Por esto, y debido a que en varias de las pruebas realizadas se encontró un estado de la red con tráfico muy cercano a cero y pocos usuarios conectados, se decidió coordinar con una maestra una actividad didáctica, sin desviarse del cronograma del curso, en la que los alumnos tuvieran que conectarse a internet y descargar algún tipo de contenido multimedia. La actividad escogida consistió en el streaming de videos de YouTube acerca de los temas vistos en clase. De esta manera se generó un determinado tráfico que serviría para poner a prueba a los modelos bajo distintos escenarios.

Pese a que los usuarios estaban descargando permanentemente video de internet, el limitado ancho de banda de la escuela hizo imposible que se dieran escenarios en los cuales alguno de los nodos de la situación inicial se encontrara saturando la celda. Se pudo estudiar entonces únicamente el caso en el que no habían usuarios saturados, con lo cual la estimación para el nuevo usuario se calcula como:

$$C_{nuevoUsuario} = C_{total} - C_{cursado}$$

En la figura 5.13 pueden verse los resultados obtenidos de la estimación de capacidad para un nuevo usuario luego de la comparación de la estimación con el tráfico efectivamente obtenido por él.

Pese a que el promedio del error relativo es muy bajo, como puede verse en la figura 5.13, estos alcanzan errores puntuales en el entorno del 20% tanto positivos como negativos. Estos resultados serán analizados en la siguiente sección.

## Capítulo 5. Segundo modelo de capacidad

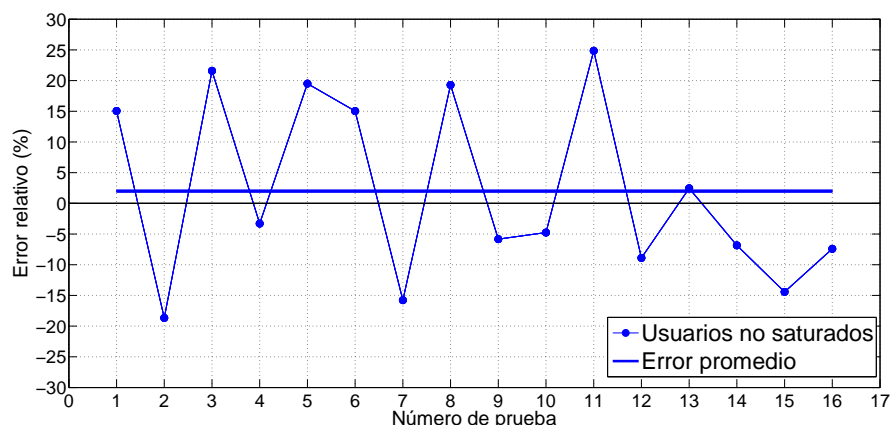


Figura 5.13: Error relativo con usuarios activos.

## 5.6. Análisis de los resultados

### 5.6.1. Escenario sin usuarios saturados

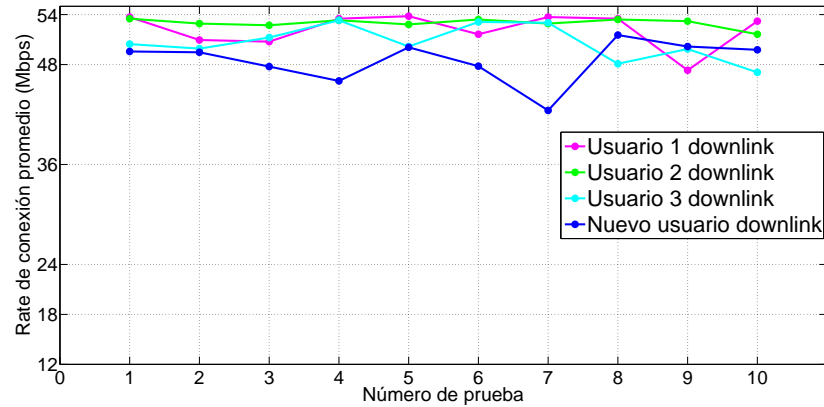
Como se ve en la figura 5.10, en las pruebas de laboratorio el modelo 2, que se basaba en la diferencia entre la capacidad total según el modelo 1 y el throughput medido, obtuvo errores relativos en el entorno del 8% y siempre por encima del throughput obtenido.

Haciendo un análisis un poco más profundo de la situación se puede obtener, mediante el uso del script *analizo\_rates.sh*, la variación de las tasas de conexión de los nodos activos previo a la entrada del nuevo usuario y compararlas con las tasas a las que este se conectó. En la figura 5.14 se grafican los promedios de las tasas de transmisión y recepción a lo largo de cada una de las pruebas para cada usuario, y las tasas de conexión del nuevo usuario una vez que ingresó a la red.

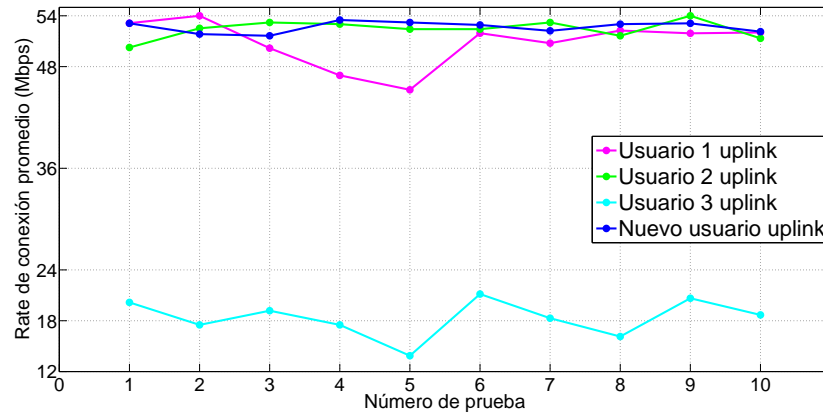
Puede notarse que el nuevo usuario tuvo en general tasas de bajada inferiores a las obtenidos por los otros usuarios conectados; además, recordando lo visto en la sección 4.5 que mostraba que las tasas de bajada son las que más influyen en el throughput total obtenido, es razonable que el nuevo usuario obtenga throughputs menores al estimado. Es decir, suponer que el nuevo usuario se comporta de la misma manera que el resto, lleva, en este caso, a estimar una capacidad un poco mayor que la que efectivamente obtiene. De todas formas, considerando la magnitud de los errores obtenidos se puede decir que el modelo, con dicha hipótesis, se comporta correctamente.

Por otra parte, durante las pruebas realizadas en la escuela se obtuvieron errores mayores que en el laboratorio y con valores tanto por encima como por debajo del throughput obtenido. Esto es en principio esperable dado que las condiciones en la escuela son mucho más variables que en un entorno de laboratorio. Repitiendo el análisis anterior, se muestran en la figura 5.15 las tasas de recepción promedio

## 5.6. Análisis de los resultados



(a) Promedios de las tasas de bajada por prueba.



(b) Promedios de las tasas de subida por prueba.

Figura 5.14: Comparación entre las tasas de conexión promedio de cada nodo con las tasas del nuevo usuario.

por prueba para cada usuario antes del ingreso del nuevo, y las tasas a las cuales este se conectó. Como se puede observar no hay una relación evidente entre sobrestimación o subestimación y el hecho de que las tasas del nuevo usuario estén por debajo o por encima del resto. Es decir, en algunas pruebas en que el nuevo usuario obtuvo tasas superiores al resto se sobrestimó su throughput, mientras que en otros casos se subestimó. Se supone esto puede deberse a la gran variabilidad que existe en las tasas de conexión de cada usuario entre dos mediciones consecutivas.

A la hora de realizar la estimación otro criterio posible es asumir que el nuevo usuario se conecta a una tasa predefinido en función de sus prestaciones. La elección de esta tasa, que será una entrada adicional al modelo, permite obtener una estimación de mejor caso posible.

## Capítulo 5. Segundo modelo de capacidad

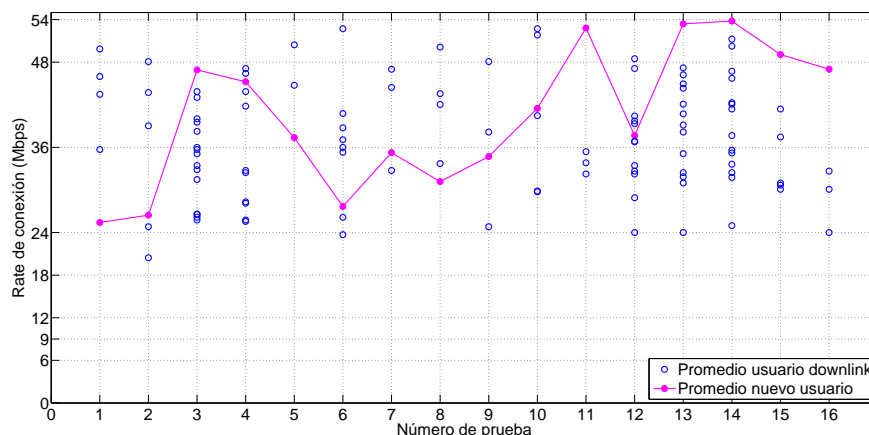


Figura 5.15: Tasas promedio por usuario por prueba, y tasas promedio nuevo usuario en escuela.

### 5.6.2. Escenario en que hay $n$ usuarios saturados

A partir de las pruebas realizadas en el laboratorio se eligió el modelo 2.iv como el más apropiado para posibles implementaciones bajo este escenario. Este modelo sugiere que al agregar un usuario saturado en una red con pocos usuarios conectados aumenta el throughput total que estos obtienen, en un cierto porcentaje de la capacidad total de red. El máximo aumento posible es de 16,7%, y se da para el caso en que hay un usuario saturado y dos o tres usuarios en total.

A continuación se analiza el comportamiento de este modelo en una de las pruebas de laboratorio. El caso más interesante de analizar es aquel en que el aumento de throughput es el mayor posible, que de acuerdo a lo ya visto se da cuando siendo  $u$  mayor que  $n$ , la diferencia entre estos es la menor posible. De las pruebas en laboratorio realizadas, la que más se asemeja a esta situación es aquella en la que al inicio hay un único usuario saturado. En esta prueba se contaba con 4 usuarios conectados por lo que el aumento de throughput debería ser 15% de la capacidad total.

En la figura 5.16 se muestra cómo aumenta el throughput obtenido por los usuarios saturados al ingresar el nuevo y se compara con el aumento estimado del 15%. Se puede observar que tras el ingreso del nuevo usuario el throughput total de los usuarios saturados aumenta, y se observa además que el valor de throughput estimado (Saturados + Aumento) es muy similar al medido luego de la entrada del nuevo usuario.

Por otro lado se puede analizar el impacto del nuevo usuario saturado sobre el throughput cursado por los usuarios no saturados. Una de las hipótesis de los modelos propuestos era que el ingreso del nuevo usuario no perjudica significativamente a los no saturados. En las figuras 5.17 y 5.18 se muestra el throughput obtenido por los usuarios no saturados antes y después del ingreso del nuevo usuario, a lo largo de diez pruebas de laboratorio, en los casos en que inicialmente

## 5.6. Análisis de los resultados

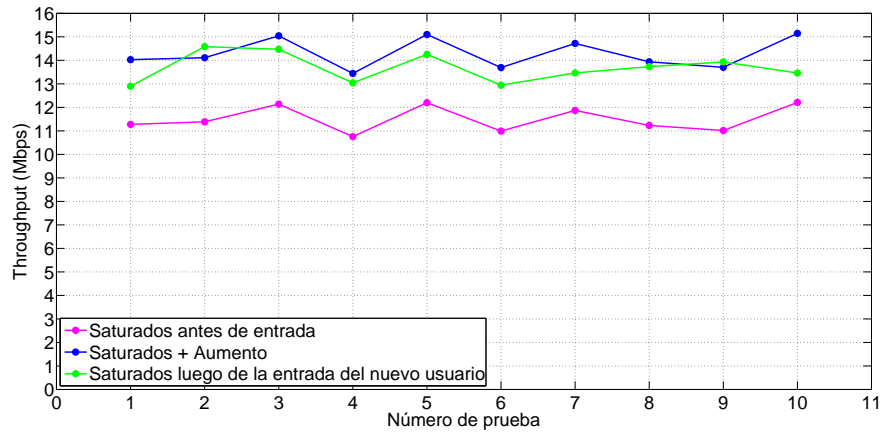


Figura 5.16: Throughput de los usuarios saturados medido y estimado.

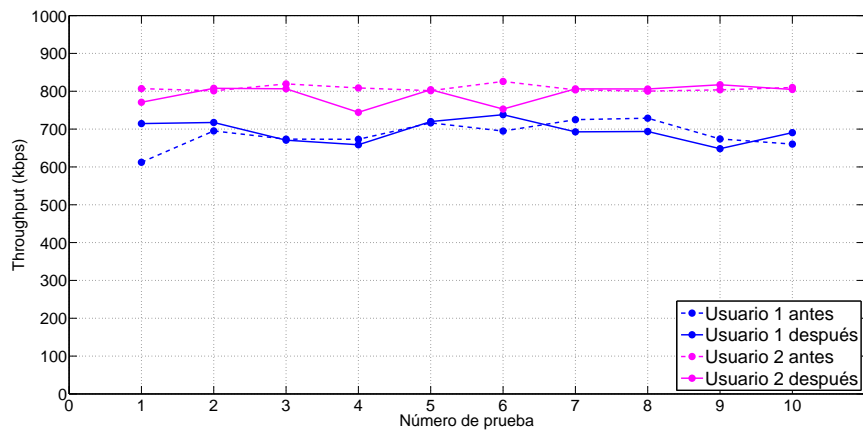


Figura 5.17: Throughput de los usuarios no saturados antes y después del ingreso del nuevo usuario. Pruebas con un saturado al inicio.

habían uno y dos usuarios saturando. Se puede ver que en ninguno de los dos casos los usuarios sufren un impacto negativo ya que el throughput obtenido por ellos no disminuye de forma sostenida, habiendo incluso aumentado en algunas pruebas. Esto es un indicador de la validez de la hipótesis considerada.

## Capítulo 5. Segundo modelo de capacidad

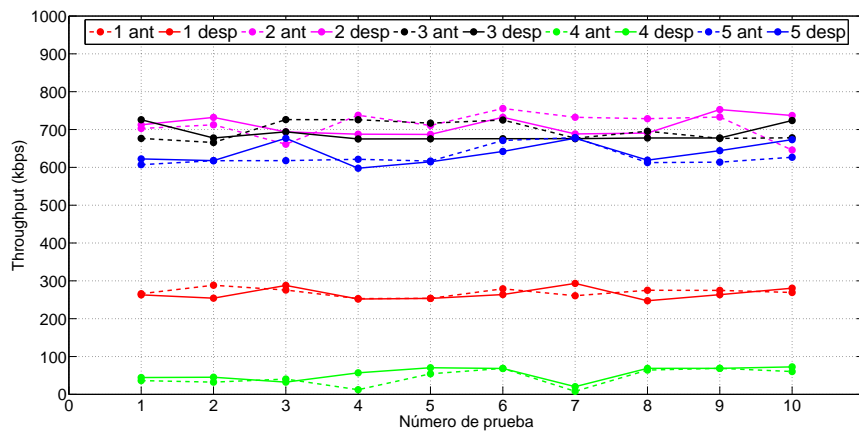


Figura 5.18: Throughput de los usuarios no saturados antes y después del ingreso del nuevo usuario. Pruebas con dos saturados al inicio.

# Capítulo 6

## Sistema de monitoreo

Tras haber escogido, en el capítulo anterior, un modelo matemático para la estimación del throughput máximo que puede sacar un nuevo usuario que se conecte a la red, se quiere ahora diseñar una herramienta que permita desplegar en tiempo real tanto esa como algunas otras métricas de interés.

Se diseña entonces una aplicación de monitoreo, accesible por web e implementada y basada en RRDtool, que grafica en tiempo real la capacidad total de la red, el throughput total medido, la estimación de throughput para un nuevo usuario y la cantidad de usuarios conectados.

En este capítulo se describe el funcionamiento de los scripts implementados así como también se describen las principales ventajas de la implementación elegida para el sistema de monitoreo.

### 6.1. Implementación

Para la implementación de cualquier sistema de monitoreo en tiempo real se deben diseñar tres grandes bloques. Estos son la obtención de los datos, su procesamiento, y el despliegue de los resultados en una interfaz que permita identificar a simple vista la evolución de una métrica a controlar o la identificación de eventos importantes.

Se implementan entonces en Shell una serie de scripts que se encargan de llevar a cabo cada una de estas funciones.

#### Bloque 1: `obtengo_datos.sh`

Basándose en la implementación de los scripts de consultas SNMP de los modelos 1 y 2, este script se inicia automáticamente al encender el servidor y consulta periódicamente al AP por los OIDs de interés. Las diferencias principales

## Capítulo 6. Sistema de monitoreo

con respecto a los scripts ya implementados radican en que este nuevo script se estará ejecutando permanentemente. Esto hace que la tarea de identificar una primera y una última consulta sea imposible, obligando a cambiar la manera en que se guardan y leen los archivos con la información obtenida. Se utilizan entonces en el nombre de los archivos marcas de tiempo de la forma HH-MM<sup>1</sup>. Por otro lado, en el bloque del procesamiento de datos será necesario contar con información obtenida acerca de los bytes transmitidos para el cálculo del throughput. Dado que nuevamente es imposible identificar una primera y última consulta SNMP, se decide por consultar por los bytes cursados en todas las consultas a diferencia de lo que se venía haciendo.

### Bloque 2: `proceso_datos.sh` y `simontraff.sh`

Esta es la etapa más compleja de todas por lo cual su implementación se realiza en dos scripts. El primero de ellos, `proceso_datos.sh`, se encarga de calcular en base a los datos obtenidos el throughput total cursado por todos los nodos de la red, la estimación de la capacidad total de la red y la capacidad para un nuevo usuario. Para esto únicamente se considera, tal como se hizo en la implementación del modelo 2, los datos obtenidos en un intervalo de tiempo de 3 minutos. En este caso serán los 3 minutos previos a su ejecución. Despliega también la cantidad de usuarios conectados al final de la prueba y la cantidad de usuarios que saturaron durante la prueba.

Para graficar estos datos y ver su evolución en el tiempo es necesario contar con alguna herramienta que permita guardar valores históricos de las muestras. Es por esto que se decide utilizar el paquete `RRDtool` (Round Robin Database Tool) para Debian que permite tanto el almacenamiento de los datos en bases de datos RRD como la posterior creación de figuras PNG con gráficas de los datos almacenados. El script `simontraff.sh` se encarga de crear dichas bases de datos y de actualizarlas periódicamente mediante la invocación de `proceso_datos.sh` y la redirección de su salida hacia ellas.

Para la actualización periódica de las bases de datos con los resultados obtenidos luego del procesamiento se recurre al uso del `cron` de los sistemas operativos Unix. Cron es básicamente un administrador de procesos en segundo plano en donde puede agendarse la ejecución de cualquier tarea a intervalos de tiempo regulares. De esta manera, configurando para que el script `simontraff.sh` se ejecute cada intervalos de 3 minutos, se actualiza la base de datos con los resultados obtenidos del procesamiento de las consultas realizadas en dicho período.

Finalmente, este script también se encarga de mantener controlado el espacio en disco ocupado por los datos obtenidos mediante `obtengo_datos.sh` simplemente

---

<sup>1</sup>HH corresponde a la hora actual en formato de 24 horas, y MM corresponde al minuto actual. Por cuestiones de implementación se omite el 0 que se antepone a las horas y minutos menores a 10.



eliminando del disco la información que ya no será procesada en posteriores ejecuciones. Es decir, cada 3 minutos procesa la información de los datos obtenidos durante dicho período, tras lo cual procede a eliminarla debido a que ya no se utilizará.

### Bloque 3: *grafsimon.sh*

Para el despliegue de los resultados se decide utilizar una interfaz web en la cual se visualiza en tiempo real la evolución del throughput medido, las capacidades estimadas para cada uno de los modelos y la cantidad usuarios conectados totales. La gran ventaja de utilizar una herramienta con interfaz web es que la aplicación se está ejecutando sobre el servidor por lo cual el sistema del gestor o administrador de la red no se verá impactado por su ejecución. Además, de esta manera también se logra una mayor compatibilidad de la aplicación sobre distintas plataformas independizándose del sistema operativo sobre el cual se está trabajando.

El script *grafsimon.sh* recibe como entrada el intervalo de tiempo hacia atrás que se desea graficar. En base a esto genera, utilizando la función *rrdtool graph* del paquete RRDtool, las figuras en formato PNG que contienen cada una de las gráficas asociadas a las distintas métricas y genera la página web que las despliega. De esta manera, accediendo en el navegador a la URL que señala a la ubicación del script en el servidor, este es ejecutado y se generan y despliegan las gráficas para el rango de tiempo seleccionado. Para ingresar al sistema se debe acceder desde cualquier navegador web a la siguiente URL:

```
<IP_servidor>/cgi-bin/grafsimon.sh?1h
```

Modificando el parámetro 1h en dicha URL se selecciona cuanto tiempo histórico desea desplegarse. Ingresando por ejemplo 30m pueden verse 30 minutos de información histórica, con 7d siete días, y con 1y un año.

Finalmente, mediante la inclusión de la etiqueta HTML:

```
<meta http-equiv="refresh" content="30" />
```

en el encabezado de la página web creada, se fuerza al navegador a actualizar dicho sitio cada 30 segundos. De esta manera se asegura que la información desplegada en determinado instante tenga retardos menores a 30 segundos, con lo cual se obtiene un sistema en tiempo real. Este tiempo puede ser decrementado a gusto del administrador a expensas de recursos consumidos en el servidor.

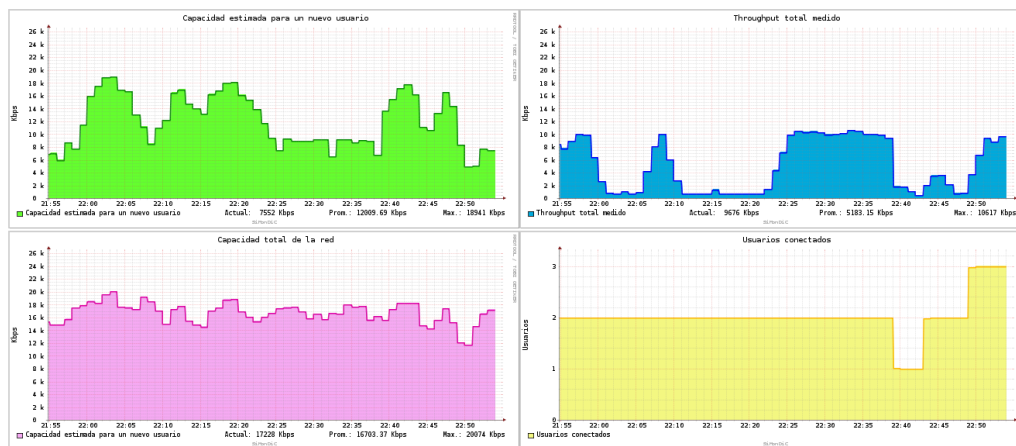
Esta implementación implica que el script es ejecutado y las gráficas son generadas únicamente cuando el administrador mantenga el sitio abierto, o cuando

## Capítulo 6. Sistema de monitoreo

acceda a él, con lo que se disminuye la carga del servidor cuando no se esté monitoreando la red.

En la figura 6.1 pueden verse algunas de las gráficas generadas por el sistema de monitoreo en donde puede verse minuto a minuto la estimación de la capacidad total, la capacidad para un nuevo usuario, el throughput medido y el número de usuarios conectados.

### SiMonDiC - Sistema de Monitoreo de Capacidad



Más graficas  
[1 hora](#) | [12 horas](#) | [1 día](#) | [7 días](#)

Figura 6.1: Interfaz de la herramienta de monitoreo web.

# Capítulo 7

## Conclusiones

### 7.1. Conclusiones generales

El modelo que estima la capacidad total de la red fue implementado satisfactoriamente logrando errores menores al 7%, lo que cumple ampliamente con el objetivo propuesto por lo que se considera un resultado muy satisfactorio.

Para la estimación del throughput que obtendrá un nuevo usuario se implementaron cuatro modelos distintos para el escenario en que la red contaba con usuarios saturados, y un modelo para el escenario en que no había ningún usuario saturado. Los errores obtenidos en el segundo caso son menores al 17%. Por su parte, en el escenario en que hay usuarios saturados, tres de los cuatro modelos propuestos se consideran válidos con errores relativos inferiores al 20% en la gran mayoría de los casos. En particular, el modelo 2.iv elegido por su mejor desempeño frente al resto obtuvo en un 90% de las pruebas errores relativos menores al 10%, cumpliendo así el objetivo propuesto. Este modelo representa además una cota superior para el throughput máximo posible ya que hace estimaciones, de forma consistente, por encima del valor de throughput que realmente obtiene el nuevo usuario.

Se implementó además una herramienta de monitoreo en tiempo real, que entendemos complementa enormemente el trabajo realizado, y representa una aplicación directa de los modelos implementados que puede ser utilizada con diversos fines.

Cabe destacar las ventajas de la solución planteada ya que utiliza un método de estimación pasivo, que no interfiere ni afecta el desempeño de la red y no requiere instalación o desarrollo de hardware adicional al ya existente. El hecho de ser simplemente software que se ejecuta sobre los servidores que se encuentran directamente conectados a los APs permite abaratar costos y tiempos de implementación de la solución.

## Capítulo 7. Conclusiones

Considerando las grandes dificultades que supone hacer una estimación sobre algo tan variable como los throughputs en redes inalámbricas, y vistas las magnitudes de los errores obtenidos y las ventajas de la solución desarrollada, se considera que el proyecto fue exitoso y logró cumplir con los objetivos propuestos.

### 7.2. Trabajos futuros

Este proyecto se basó en las revisiones b y g del estándar 802.11 debido a que son las más utilizadas en las redes de Plan Ceibal. Sin embargo, en la actualidad existen revisiones que permiten mayores velocidades de transmisión tales como las revisiones n y ac. Estas hacen uso de la agregación de tramas a nivel de capa MAC de manera de transmitir un solo encabezado por varios paquetes de información. A su vez es usual que las redes se configuren en modo mixto permitiendo la coexistencia de dispositivos que trabajan en diferentes revisiones del estándar. Se plantea entonces como posible trabajo a futuro investigar el comportamiento de las redes en nuevas revisiones del estándar 802.11 y/o en modo mixto, y adaptar las implementaciones aquí desarrolladas a dichos escenarios.

Otra posible ampliación de este proyecto podría ser el desarrollo de aplicaciones que tomen decisiones en función de la información obtenida a partir de los modelos aquí desarrollados, como por ejemplo el control de acceso a usuarios. A su vez, en casos en que se detecte que la capacidad de la red es muy baja, teniendo en cuenta el fenómeno de degradación debido a multi-rate, podría implementarse un software capaz de cambiar los parámetros de configuración de la red evitando que se conecten usuarios a tasas bajas.

El software fue desarrollado en Shell, que a pesar de cubrir ampliamente las necesidades de este proyecto, requiere lectura y escritura permanente de archivos en disco. Esto aumenta el tiempo de ejecución y la carga del servidor cuando aumenta considerablemente la cantidad de usuarios conectados a la red. Se propone entonces la implementación de los modelos en otro lenguaje de programación con manejo más eficiente de datos, o utilizando bases de datos.

# Apéndice A

## Degradación del desempeño debido al multi-rate

En una red inalámbrica las condiciones del medio pueden variar significativamente debido a factores tales como interferencias de otras redes inalámbricas, señales de otros dispositivos en la misma banda de frecuencia o por el simple movimiento de los clientes, provocando esto posibles pérdidas de paquetes. Para lidiar con este problema, el mecanismo ARF adapta las tasas de conexión de los nodos de la red de manera de hacer más robustas las comunicaciones entre ellos. Otro motivo de que existan nodos conectados a distintas tasas es la coexistencia de diferentes versiones del estándar 802.11 en la misma red. Debido al funcionamiento del protocolo de acceso al medio (CSMA/CA) la coexistencia de nodos a tasas bajas y nodos a tasas altas compitiendo por el canal disminuirá las velocidades máximas de transmisión de todos los nodos conectados.

En este apéndice se describe y analiza, profundizando el análisis temporal visto en la sección 4.2, el efecto de degradación de desempeño de los nodos en una red inalámbrica cuando en esta hay dispositivos conectados a tasas altas y dispositivos conectados a tasas bajas. El análisis aquí presentado se basa en lo presentado en [27] para el estándar 802.11b pero los resultados pueden extenderse a cualquier revisión del estándar 802.11.

### A.1. Análisis

Como se vio en la sección 4.2 el tiempo de transmisión de una trama para un nodo cualquiera de la red es:

$$T_{total} = T_{data} + T_{ack} + T_{backoff}$$

donde:

$$T_{data} = DIFS + SIFS + T_{802,11-data} + T_{802,11-ack}$$

## Apéndice A. Degradación del desempeño debido al multi-rate

$$T_{ack} = DIFS + SIFS + T_{tcp-ack} + T_{802,11-ack}$$

$$T_{backoff} = \frac{aSlotTime(W_{min} - 1)}{2}$$

Para 802.11g se tiene:

$$T_{802,11-data} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{802,11_{data}}{Tx_{rate}} + T_{signal-extension}$$

$$T_{802,11-ack} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{802,11_{ack}}{B_{rate}} + T_{signal-extension}$$

$$T_{tcp-ack} = \frac{PLCP_{header}}{B_{rate}} + \frac{PLCP_{preamble}}{B_{rate}} + \frac{TCP_{ack}}{Rx_{rate}} + T_{signal-extension}$$

Aquí  $B_{rate}$  es el Basic Rate configurado en el AP,  $PLCP_{header}$  es el tamaño del encabezado de PLCP,  $PLCP_{preamble}$  es el tamaño del preámbulo de PLCP,  $Rx_{rate}$  es la tasa de recepción del AP para ese nodo,  $Tx_{rate}$  es la tasa de transmisión del AP para ese nodo, y  $T_{signal-extension}$  es un tiempo adicional de 6  $\mu s$ .

Consideremos el caso en el que en la red hay  $(N - 1)$  nodos conectados a tasa  $Tx_{rate} = Rx_{rate} = R$ , y 1 nodo conectado a tasa  $Tx_{rate} = Rx_{rate} = r$ , donde  $R > r$ . Los tiempos de transmisión de una trama para los nodos de alta y baja tasa pueden escribirse como:

$$T_{total}(R) = \tau + \frac{802,11_{data} + TCP_{ack}}{R} = \tau + t_{tr}(R)$$

$$T_{total}(r) = \tau + \frac{802,11_{data} + TCP_{ack}}{r} = \tau + t_{tr}(r)$$

donde  $t_{tr}$  es el tiempo de transmisión de datos útiles y  $\tau$  es una constante de tiempo que depende de  $B_{rate}$ ,  $PLCP_{header}$ ,  $PLCP_{preamble}$ ,  $T_{signal-extension}$ ,  $aSlotTime$  y  $W_{min}$ .

Sea  $p$  la proporción de tráfico útil cursado por cada nodo se tiene:

$$p = \frac{thr_{eff}}{thr_{total}} = \frac{t_{tr}}{T_{total}} \frac{Bytes_{utiles}}{Bytes_{totales}}$$

Debido a que el mecanismo CSMA/CA otorga equiprobabilidad de acceso al medio, se puede hacer un análisis a largo plazo de la situación, asumiendo que las estaciones se alternan las transmisiones. Así, definiendo la utilización del canal por un nodo como la relación entre el tiempo que le lleva transmitir un paquete y el tiempo en el cual todos los nodos transmiten un paquete se tiene:

$$U_R = \frac{T_{total}(R)}{T_{total}(R)(N-1) + T_{total}(r)}$$

$$U_r = \frac{T_{total}(r)}{T_{total}(R)(N-1) + T_{total}(r)}$$

donde  $U_R$  y  $U_r$  representan la utilización del canal por un nodo rápido y uno lento respectivamente.

Luego, el throughput obtenido por cada uno de los nodos rápidos será:

$$thr_R = U_R \times p_R \times R$$

y el throughput obtenido por cada uno de los nodos lentos:

$$thr_r = U_r \times p_r \times r$$

Sustituyendo  $U_r$  y  $p_r$  por sus expresiones completas se obtiene:

$$thr_R = \frac{T_{total}(R)}{T_{total}(R)(N-1) + T_{total}(r)} \times \frac{802,11_{data} + TCP_{ack}}{T_{total}(R) \times R} \frac{Bytes_{utiles}}{Bytes_{totales}} \times R$$

$$thr_r = \frac{T_{total}(r)}{T_{total}(R)(N-1) + T_{total}(r)} \times \frac{802,11_{data} + TCP_{ack}}{T_{total}(r) \times r} \frac{Bytes_{utiles}}{Bytes_{totales}} \times r$$

de donde puede verse que:

$$thr_R = thr_r = \frac{802,11_{data} + TCP_{ack}}{T_{total}(R)(N-1) + T_{total}(r)} \times \frac{Bytes_{utiles}}{Bytes_{totales}}$$

Este resultado muestra que cuando un nodo conectado a tasa baja compite por el acceso al medio con nodos conectados a tasa alta, el throughput de todos los nodos se verá considerablemente limitado - los nodos rápidos ven su throughput degradado al nivel de los nodos lentos. La equiprobabilidad del acceso al medio que otorga CSMA/CA provoca que los nodos lentos capturen el canal por tiempos mucho mayores que los nodos rápidos. Es decir, un nodo transmitiendo a 9 Mbps capturará el canal por un tiempo 6 veces mayor que un nodo transmitiendo a 54 Mbps. Esta anomalía degrada el desempeño percibido por los usuarios en la celda, y es independiente de la cantidad de nodos lentos.

## Apéndice A. Degradación del desempeño debido al multi-rate

Para eliminar dicha anomalía se presentan en [28] y [29] mecanismos que modifican el tamaño de las tramas a transmitir según las tasas de conexión, cambiando el tipo de justicia, es decir, en lugar de buscar igual chance de acceso al medio se busca igual tiempo de aire. Otorgando a todos los nodos el mismo tiempo de transmisión, en definitiva provocaría que el throughput que obtenga cada nodo dependa únicamente de su tasa de transmisión.



# Apéndice B

## Gestión de proyecto

En la figura B.1 se puede ver el diagrama de Gantt elaborado al inicio del proyecto. No se identificaron grandes riesgos que pudieran perjudicar o atrasar la fecha de finalización del proyecto significativamente.

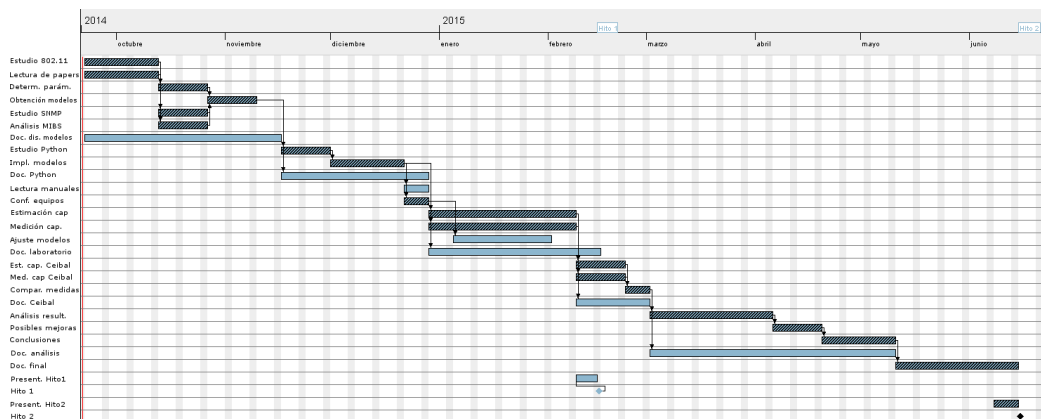


Figura B.1: Diagrama de Gantt al inicio del proyecto.

A medida que se fue avanzando con las tareas y conociendo mejor las dificultades que el proyecto presentaba, se observó que algunas tareas consumían más tiempo del previsto al inicio, y que a su vez era conveniente modificar el orden de algunas. Se aprovecharon entonces las instancias de los hitos 1 y 2 para realizar las modificaciones pertinentes a la planificación. En las figuras B.2 y B.3 se muestran los diagramas de Gantt asociados a dichas replanificaciones.

## Apéndice B. Gestión de proyecto

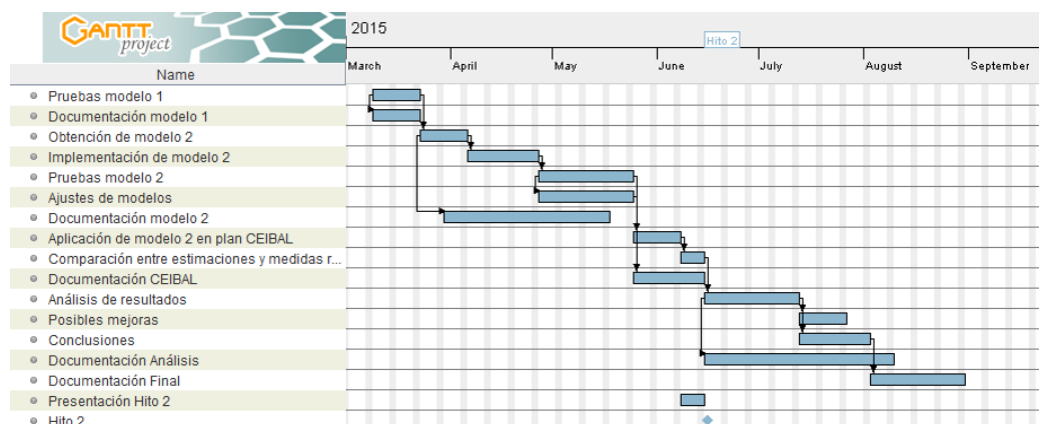


Figura B.2: Replanificación previa al Hito 1.

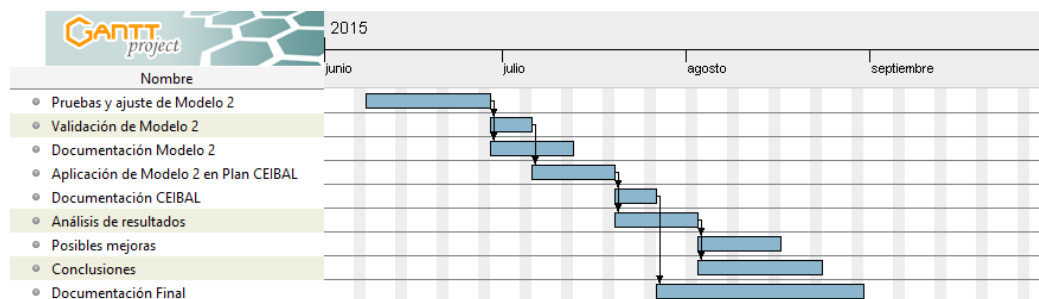


Figura B.3: Replanificación previa al Hito 2.

Pese a las replanificaciones realizadas, debido a los paros de enseñanza ocurridos durante los meses de agosto y setiembre del 2015, fue necesario solicitar un mes de prórroga para finalizar el proyecto. Este riesgo no fue tomado en cuenta durante el análisis de riesgos al inicio, y su impacto fue importante ya que imposibilitó por un largo período de tiempo la ida a la escuela para la validación de los modelos en campo. Debido a que este fue un año de asignación de presupuestos por parte del gobierno, podía ser previsible un problema de este tipo.

# Apéndice C

## Contenido del CD

En la figura C.1 se puede ver la estructura de carpetas del CD. Esta se detalla a continuación.

- **Modelo 1 y Modelo 2:** se encuentran aquí todos los scripts desarrollados para la implementación de los dos modelos de capacidad, así como también los scripts desarrollados para el reprocesamiento de los datos, la elección del tiempo de medición, y el análisis de las tasas de transmisión. Se encuentran aquí también los siguientes directorios:
  - **grafica:** contiene los siguientes directorios:
    - **datos:** contiene todos los datos obtenidos.
    - **para\_graficar:** contiene los datos a graficar.
    - **tmp:** carpeta de archivos temporales.
  - **resultados:** carpeta donde se almacenan salidas de la ejecución automática del script del Modelo 1.
  - **salidas:** carpeta donde se almacenan las salidas de la ejecución actual.
  - **salidas\_historicas:** carpeta donde se almacenan todas las salidas históricas para su posterior análisis y/o reprocesamiento.
  - **salidas\_rates:** salidas del script de análisis de tasas de transmisión.
  - **salidas\_tiempos:** salidas del script de elección de tiempo de medición.
  - **tmp:** carpeta de archivos temporales.
- **Sistema de monitoreo:** se encuentran aquí los cuatro scripts desarrollados para la implementación del sistema de monitoreo.

Apéndice C. Contenido del CD

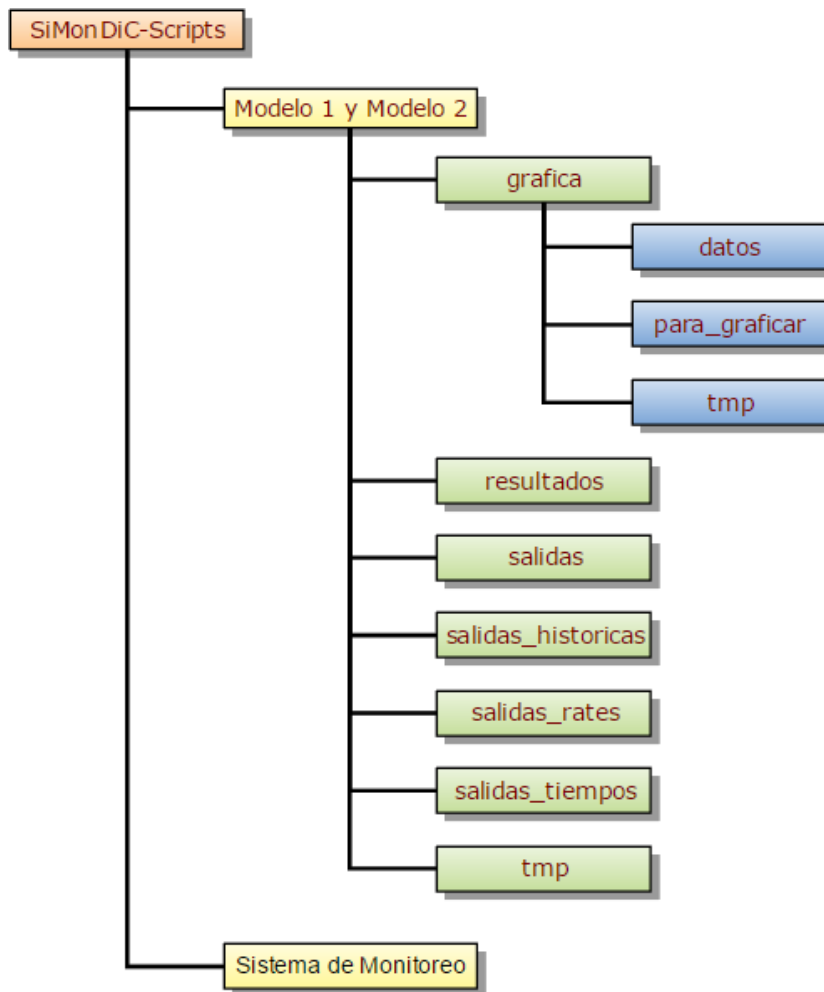


Figura C.1: Estructura de carpetas del CD.

# Referencias

- [1] IBM Rueschlikon Labs. Diffused Infrared WLAN. 1979.
- [2] 802.11-2012 - IEEE Standard for Information technology– Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,. 2012.
- [3] Plan Ceibal. <http://www.ceibal.edu.uy/>.
- [4] Federico La Rocca and Fernanda Rodríguez. An Overview of WLAN Performance, Some Important Case-Scenarios and their Associated Models. *Wireless Personal Communications*, 79:131–184, November 2014.
- [5] Raffaele Bruno, Marco Conti, and Enrico Gregori. Throughput Analysis and Measurements in IEEE 802.11 WLANs with TCP and UDP Traffic Flows. *IEEE Transactions on Mobile Computing*, 7(2), February 2008.
- [6] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal*, 18(3):535–547, 2000.
- [7] Sunwoong Choi, Kihong Park, and Chong kwon Kim. On the Performance Characteristics of WLANs: Revisited. In *in Proc. ACM SIGMETRICS'05*, pages 97–108, 2005.
- [8] Samarth H. Shah, Kai Chen, and Klara Nahrstedt. Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks. *Proceedings of 1st ISMA/CAIDA Workshop on Bandwidth Estimation (BEst)*, December 1997.
- [9] Yi Ding. *A Remote Capacity Utilization Estimator for WLANs*. PhD thesis, Dublin Institute of Technology, 2014.
- [10] Vinton G. Cerf and Robert E. Kahn. A Protocol for Packet Network Intercommunication. *Communications, IEEE Transactions on*, 22, May 1974.
- [11] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *TCP Selective Acknowledgment Options*, October 1996.
- [12] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. *Introduction to Community-based SNMPv2*, January 1996.

## Referencias

- [13] K. McCloghrie and M. Rose. *Management Information Base for Network Management of TCP/IP-based Internets*, March 1991.
- [14] R. Presuhn, J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*, December 2002.
- [15] Devin Akin and Jim Geier. *CWAP Certified Wireless Analysis Professional Official Study Guide*. Brandon A. Nordin, 2004.
- [16] Ad Kamerman and Leo Monteban. WaveLAN-II: A High-performance wireless LAN for the unlicensed band. *Bell Lab Technical Journal*, pages 118–133, 1997.
- [17] Plan Ceibal - Especificaciones técnicas. <http://www.ceibal.edu.uy/art%C3%ADculo/noticias/estudiantes/Especificaciones-tecnicas/>.
- [18] Routerboard RB433AH Product Specifications. <http://routerboard.com/RB433AH/>.
- [19] Routerboard R52N Product Specifications. <http://routerboard.com/R52N/>.
- [20] iPerf - The network bandwidth measurement tool. <https://iperf.fr/>.
- [21] MikroTik software products. <http://www.mikrotik.com/download/>.
- [22] VirtualBox. <https://www.virtualbox.org/>.
- [23] WinSCP - Free SFTP, SCP and FTP client for Windows. <https://winscp.net/eng/docs/lang:es/>.
- [24] ManageEngine - Free SNMP MIB Browser. <https://www.manageengine.com/products/mibbrowser-free-tool/>.
- [25] RRDtool logging and graphing. <http://oss.oetiker.ch/rrdtool/>.
- [26] IEEE. IEEE 802.11g-2003: Further Higher Data Rate Extension in the 2.4 GHz Band. 2003.
- [27] Martin Heusse, Franck Rousseau, Gilles Berger-Sabbatel, and Andrzej Duda. Performance Anomaly of 802.11b. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2:836–843, 2003.
- [28] Osama Abu-Sharkh and Ahmed H. Tewfik. Throughput evaluation and enhancement in 802.11 WLANs with access point. *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 2:1338–1341, 2005.

- [29] Ilenia Tinnirello and Sunghyun Choi. Temporal Fairness Provisioning in Multi-Rate Contention-Based 802.11e WLANs. *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*, pages 220–230, 2005.

Esta página ha sido intencionalmente dejada en blanco.



# Índice de tablas

2.1. Parámetros de los estándares 802.11b y 802.11g. . . . .	16
4.1. Parámetros de los estándares 802.11b y 802.11g . . . . .	28
4.2. Parámetros de los estándares 802.11b y 802.11g . . . . .	28
5.1. Valores de throughput estimado para el nuevo usuario cuando no hay usuarios conectados a la red. . . . .	45
5.2. Errores relativos y desviación estándar de los modelos 2.ii, 2.iii y 2.iv con n la cantidad de usuarios saturados al inicio de la prueba .	61
5.3. Porcentaje de estimaciones por encima del valor obtenido . . . . .	61

Esta página ha sido intencionalmente dejada en blanco.

# Índice de figuras

1.1. Modelo TCP/IP. . . . .	3
2.1. Paquete TCP. . . . .	8
2.2. Datagrama UDP. . . . .	9
2.4. Composición de la trama PDU. Imagen extraída de [15]. . . . .	11
2.3. Ejemplo de una MIB cargada con un explorador de MIBs en donde puede verse la estructura tipo árbol. El OID seleccionado corresponde al de los bytes transmitidos hacia cada nodo conectado. . . . .	12
2.5. Diagrama de flujo de CSMA/CA. . . . .	15
2.6. Ejemplo de diagrama temporal del mecanismo CSMA/CA en el que tres nodos siempre tienen un paquete listo para transmitir. . . . .	16
3.1. Diagrama de la red implementada. . . . .	20
3.2. Modelos de laptops del Plan Ceibal utilizados. . . . .	20
3.3. Red de laboratorio . . . . .	21
3.4. Configuración de las interfaces inalámbricas en Winbox. . . . .	22
3.5. Ejemplo de gráfica generada con RRDtool: tráfico por una interfaz. . . . .	24
4.1. Diagrama del script <i>consultaSNMP_rate.sh</i> . . . . .	30
4.2. Diagrama del script <i>llamo_consultaSNMP_rate.sh</i> . . . . .	33
4.3. Throughput de la red según mediciones de distinta duración. . . . .	34
4.4. Desviación estándar del throughput para cada intervalo de medición. . . . .	35
4.5. Reparto de throughput para 3 usuarios conectados. . . . .	36
4.6. Tasas de bajada en una prueba de laboratorio con 4 usuarios conectados. . . . .	36
4.7. Tasas de subida en una prueba de laboratorio con 4 usuarios conectados. . . . .	36
4.8. Variación de tasas de conexión de bajada y subida y throughput estimado para un usuario particular. . . . .	37
4.9. Resultados modelo 1: 1 usuario. . . . .	38
4.10. Resultados modelo 1: 2 usuarios. . . . .	38
4.11. Resultados modelo 1: 3 usuarios. . . . .	38
4.12. Resultados modelo 1: 4 usuarios. . . . .	38
4.13. Resultados modelo 1: 5 usuarios. . . . .	38
4.14. Resultados modelo 1: 6 usuarios. . . . .	38
4.15. Resultados modelo 1: 7 usuarios. . . . .	39

## Índice de figuras

4.16. Resultados modelo 1: 8 usuarios. . . . .	39
4.17. Resultados modelo 1: 9 usuarios. . . . .	39
4.18. Throughput promedio medido y estimado para distintas cantidades de usuarios conectados. . . . .	40
4.19. Error relativo del throughput estimado promedio respecto al medido promedio. . . . .	40
5.1. Proceso de iteración para el cálculo del umbral de saturación. En verde se indican los usuarios clasificados como saturados en cada iteración. . . . .	43
5.2. Reparto de throughput según modelo 2.i. . . . .	45
5.3. Reparto de throughput según modelo 2.iv. . . . .	48
5.4. Diagrama de tráfico por usuario, antes y después del ingreso del nuevo usuario saturado. . . . .	49
5.5. Aumento de throughput de los usuarios saturados al ingresar uno nuevo. . . . .	51
5.6. Diagrama del script <i>modelo2-consultaSNMP_rate.sh</i> . . . . .	52
5.7. Distintos casos de desconexiones temporales. En verde se indican los intervalos de tiempo en los que el nodo estaba conectado. . . .	54
5.8. Ejemplo de salida del script <i>procesamiento.sh</i> . . . . .	56
5.9. Ejemplo de salida del script <i>analizo_rates.sh</i> . . . . .	56
5.10. Resultados obtenidos aplicando el modelo 2 en el laboratorio para el caso en que no hay usuarios saturados. . . . .	59
5.11. Comparación de los errores relativos obtenidos por los modelos propuestos. . . . .	60
5.12. Variación de clientes conectados y tráfico de downlink para el AP. . .	62
5.13. Error relativo con usuarios activos. . . . .	64
5.14. Comparación entre las tasas de conexión promedio de cada nodo con las tasas del nuevo usuario. . . . .	65
5.15. Tasas promedio por usuario por prueba, y tasas promedio nuevo usuario en escuela. . . . .	66
5.16. Throughput de los usuarios saturados medido y estimado. . . . .	67
5.17. Throughput de los usuarios no saturados antes y después del ingreso del nuevo usuario. Pruebas con un saturado al inicio. . . . .	67
5.18. Throughput de los usuarios no saturados antes y después del ingreso del nuevo usuario. Pruebas con dos saturados al inicio. . . . .	68
6.1. Interfaz de la herramienta de monitoreo web. . . . .	72
B.1. Diagrama de Gantt al inicio del proyecto. . . . .	79
B.2. Replanificación previa al Hito 1. . . . .	80
B.3. Replanificación previa al Hito 2. . . . .	80
C.1. Estructura de carpetas del CD. . . . .	82



Esta es la última página.  
Compilado el lunes 7 marzo, 2016.  
<http://iie.fing.edu.uy/>