



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# Mediciones y Aprendizaje Estadístico aplicado a problemas de Redes Inalámbricas

TESIS PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA  
UNIVERSIDAD DE LA REPÚBLICA POR

Claudina Rattaro

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS  
PARA LA OBTENCIÓN DEL TÍTULO DE  
MAGISTER EN INGENIERÍA ELÉCTRICA.

DIRECTOR DE TESIS

Dr. Ing. Pablo Belzarena ..... Universidad de la República

TRIBUNAL

Msc. Ing. Gabriel Gómez ..... Universidad de la República

Dr. Ing. Andrés Ferragut ..... Universidad ORT

Dr. Ing. Rafael Sotelo ..... Universidad de la República

DIRECTOR ACADÉMICO

Dr. Ing. Pablo Belzarena ..... Universidad de la República

Montevideo  
19/10/2012

*Mediciones y Aprendizaje Estadístico aplicado a problemas de Redes Inalámbricas,*  
Claudina Rattaro

ISSN 1688-2806

Esta tesis fue preparada en L<sup>A</sup>T<sub>E</sub>X usando la clase iietesis (v1.0).

Contiene un total de 121 páginas.

Compilada el 19 de noviembre de 2012.

<http://iie.fing.edu.uy/>

# Agradecimientos

Quiero agradecer en primer lugar a mi director de tesis por haber confiado en mi, por todo el tiempo dedicado y por su orientación necesaria para el desarrollo de este trabajo.

Al Plan Ceibal por todo el apoyo brindado.

A la gente de la Universidad del Rey Juan Carlos - Madrid, especialmente al Ing. Iván Hernández y al Dr. Javier Simó, quienes trabajaron conmigo durante la pasantía en dicha institución.

Al apoyo de la Red MASERATTI (Mejora de la Atención Sanitaria en Entornos Rurales mediante Aplicaciones de Telemedicina sobre Tecnologías Inalámbricas) financiada por el programa CYTED.

Al proyecto CSIC “Algoritmos de Control de Acceso al Medio en Redes Inalámbricas” en especial al Dr. Federico Larroca por todos sus aportes y consejos.

Al Cluster Fing y a Gabriel Gómez por ayudarme a usarlo.

A mis amigos y compañeros de trabajo (en particular a Cesó, a Fernanda y a los gurises de Ceibal).

A mi familia por el apoyo todo este tiempo.

Esta página ha sido intencionalmente dejada en blanco.

*A Marco.*

Esta página ha sido intencionalmente dejada en blanco.

# Resumen

El presente documento contiene la documentación final de la Tesis de Maestría titulada: “Mediciones y Aprendizaje Estadístico aplicado a problemas de Redes Inalámbricas”. La autora del mismo, Claudina Rattaro, es estudiante de la Maestría en Ingeniería Eléctrica de la Facultad de Ingeniería - UdelaR, Plan de Estudios 2004. El trabajo se llevó a cabo en el período comprendido entre agosto 2010 y setiembre 2012 bajo la tutoría del Doctor Pablo Belzarena.

Desde hace unos años, las redes inalámbricas se han convertido en la tecnología más popular para brindar acceso a Internet. Como consecuencia de su fácil despliegue y bajos costos, éstas juegan un papel fundamental a la hora de suministrar conectividad en entornos rurales aislados con baja densidad de población. En estas zonas es importante llegar con servicios de telemedicina y teleeducación, imprescindibles para lograr igualdad de oportunidades. Para desplegar este tipo de aplicaciones u otros servicios interactivos y audiovisuales, se necesita una infraestructura de red que garantice ciertos niveles de calidad de servicio. Las tecnologías más populares actualmente (ej.: IEEE 802.11) no pueden garantizar estrictos requisitos de calidad de servicio, por lo tanto, resulta necesaria la investigación de mecanismos adicionales.

Este trabajo propone un enfoque de “caja negra” para estudiar el desempeño de las redes inalámbricas. En particular se presenta una metodología que permite la estimación de parámetros de calidad de servicio de extremo a extremo para una conexión en una red de este tipo. La técnica propuesta está basada en una combinación de mediciones activas en la red junto con la aplicación de técnicas de aprendizaje estadístico, donde el sistema es entrenado durante ciertos períodos de tiempo mediante flujos de aplicación y el envío de ráfagas de paquetes de medida. El análisis estadístico se basa en la utilización de la herramienta Support Vector Machines. Es importante destacar que el método propuesto resulta independiente de la tecnología inalámbrica utilizada.

La metodología puede ser usada para varios propósitos. En este trabajo se presentan dos posibles aplicaciones: (i) estimación del throughput obtenido por cierta conexión en una red inalámbrica, (ii) elaboración de decisiones de control de admisión con el objetivo de garantizar requerimientos mínimos de calidad de servicio de extremo a extremo según cada aplicación.

Se realizaron validaciones por medio de simulaciones y también de forma experimental en redes montadas en un laboratorio. Se incluyeron pruebas en redes heterogéneas: backbone WiMax y tramo de acceso WiFi. Se obtuvieron buenos resultados en la verificación de ambos casos de uso de la metodología. Se demuestra además que la técnica no resulta invasiva a pesar de que involucra mediciones activas en la red.

**Palabras clave:** Aprendizaje estadístico, Support Vector Machines, Redes Inalámbricas, Control de Admisión, IEEE 802.11, IEEE 802.11e, WiMax, QoS.

# Tabla de contenidos

<b>Agradecimientos</b>	<b>I</b>
<b>Resumen</b>	<b>v</b>
<b>I INTRODUCCIÓN</b>	<b>1</b>
<b>1. Presentación</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.1.1. Problemática y necesidad . . . . .	3
1.1.2. Tecnologías de interés y sus limitaciones . . . . .	4
1.1.3. Desafíos . . . . .	6
1.2. Síntesis de la Propuesta . . . . .	7
1.3. Principales Resultados . . . . .	8
1.4. Organización del documento . . . . .	9
<b>2. Marco Teórico y Estado del Arte</b>	<b>11</b>
2.1. Calidad de Servicio en redes 802.11e . . . . .	11
2.1.1. Introducción al estándar 802.11 . . . . .	11
2.1.2. IEEE 802.11e . . . . .	13
2.2. Control de Admisión en redes 802.11e . . . . .	14
2.3. Calidad de servicio en WiMax . . . . .	16
2.3.1. Introducción . . . . .	16
2.3.2. Calidad de servicio . . . . .	17
2.4. Antecedentes de trabajos relacionados con la estimación de parámetros de QoS de extremo a extremo en redes inalámbricas . . . . .	19
2.4.1. Introducción . . . . .	19
2.4.2. Revisión de algunas propuestas en el área de modelos analíticos	20
2.4.3. Revisión de algunas propuestas en el área de mediciones . .	21
<b>3. Support Vector Machines</b>	<b>23</b>
3.1. Introducción . . . . .	23
3.2. Regresión con SVM (SVR) . . . . .	24
3.3. Clasificación con SVM (SVM-C) . . . . .	26
3.4. Implementación de SVM elegida . . . . .	26

<b>II</b>	<b>DETALLE DE LA PROPUESTA</b>	<b>29</b>
<b>4.</b>	<b>Metodología</b>	<b>31</b>
4.1.	Introducción . . . . .	31
4.2.	Estimación de parámetros de QoS . . . . .	33
4.3.	Herramientas para Decisiones de Control de Admisión . . . . .	34
<b>5.</b>	<b>Estimador del estado de la red</b>	<b>37</b>
5.1.	Introducción . . . . .	37
5.2.	Definición de $X$ . . . . .	37
<b>III</b>	<b>VALIDACIÓN DE LA METODOLOGÍA</b>	<b>43</b>
<b>6.</b>	<b>Simulaciones</b>	<b>45</b>
6.1.	Introducción . . . . .	45
6.2.	Simulador ns-2 . . . . .	45
6.2.1.	Implementación de 802.11 . . . . .	46
6.2.2.	Extensión a 802.11e . . . . .	46
6.3.	Estimación de Throughput en una red 802.11 . . . . .	48
6.3.1.	Introducción . . . . .	48
6.3.2.	Condiciones Saturadas de Tráfico . . . . .	49
6.3.3.	Condiciones NO Saturadas de Tráfico . . . . .	52
6.4.	Herramientas para decisiones de Control de Admisión para una red 802.11e . . . . .	56
6.4.1.	Introducción . . . . .	56
6.4.2.	Topología I . . . . .	58
6.4.3.	Topología II . . . . .	61
6.4.4.	Decisiones de Control de Admisión considerando no afectar a los flujos presentes en la red con el ingreso de la nueva conexión . . . . .	63
<b>7.</b>	<b>Validación Experimental considerando 802.11e y WiMAX</b>	<b>67</b>
7.1.	Introducción . . . . .	67
7.2.	Descripción del escenario de trabajo . . . . .	68
7.2.1.	Equipamiento WiMAX . . . . .	68
7.2.2.	Equipamiento WiFi . . . . .	69
7.2.3.	Equipamiento para Servidores . . . . .	70
7.2.4.	Sincronización de Relojes . . . . .	70
7.2.5.	Elección de Frecuencias y Rate Capa Física . . . . .	71
7.2.6.	Inyector de Tráfico: D-ITG . . . . .	71
7.2.7.	Configuraciones Básicas . . . . .	72
7.3.	Configuración de QoS y Verificación . . . . .	72
7.3.1.	Configuración y verificación del segmento EDCA . . . . .	72
7.3.2.	Configuración y verificación del segmento WiMAX . . . . .	74
7.4.	Automatización de las pruebas y generalidades . . . . .	75

7.5. Realización de Medidas . . . . .	76
7.6. Análisis de Resultados . . . . .	77
<b>IV CONCLUSIONES Y TRABAJO FUTURO</b>	<b>81</b>
<b>8. Conclusiones y Trabajo Futuro</b>	<b>83</b>
8.1. Conclusiones . . . . .	83
8.2. Trabajo Futuro . . . . .	83
<b>V APÉNDICES</b>	<b>85</b>
<b>A. Scripts utilizados en Pruebas de Laboratorio</b>	<b>87</b>
<b>B. Scripts utilizados en las simulaciones de ns-2</b>	<b>91</b>
<b>Referencias</b>	<b>97</b>
<b>Índice de tablas</b>	<b>103</b>
<b>Índice de figuras</b>	<b>104</b>

Esta página ha sido intencionalmente dejada en blanco.

Parte I  
INTRODUCCIÓN



# Capítulo 1

## Presentación

### 1.1. Motivación

#### 1.1.1. Problemática y necesidad

La conectividad a Internet en zonas rurales con baja densidad de población presenta algunas dificultades por el alto costo de las soluciones tradicionales de acceso para largas distancias. Este factor afecta muy negativamente diversos aspectos de la vida en estas regiones como ser la atención sanitaria o las posibilidades de educación a niños, adolescentes y adultos de dichas poblaciones. Por ejemplo, a través de estudios realizados por la Fundación EHAS [1] se ha comprobado que en las zonas rurales más aisladas existen importantes problemas en la prevención y diagnóstico prematuro de enfermedades. Experiencias previas han demostrado que algunas tecnologías inalámbricas pueden resultar adecuadas para brindar servicios básicos en estos contextos. Por otro lado tenemos que los fabricantes de tecnología vienen desarrollando equipos para realizar tareas de telemedicina y teleeducación (ej.: equipamientos de videoconferencia [2] [3]), lo que motiva a la investigación de contar con redes adecuadas para este tipo de servicios.

Existen varias experiencias piloto destacables en el mundo en cuanto a la aplicación de telemedicina y teleeducación en poblaciones rurales aisladas. En América Latina hay casos interesantes por ejemplo en Colombia, Ecuador y Perú entre otros ([4] [5]). Fuera de América Latina, India ha realizado varias experiencias (ver por ejemplo “Aravind Eye Care System” [6]). El problema principal para desplegar diversas aplicaciones de telemedicina o teleeducación u otros servicios interactivos y audiovisuales que pueden tener un impacto socio-económico potencialmente alto, es que se requiere de una infraestructura de telecomunicaciones de bajo costo, fácil mantenimiento y sobre todo con garantías de Calidad de Servicio (QoS), cosa que hasta ahora no se tiene. Para lograrlo se precisa aún más I+D tanto en las tecnologías inalámbricas elegibles como en las propias aplicaciones.

Estos problemas tienen además en Uruguay, un soporte y un impacto claro en

## Capítulo 1. Presentación

la red del Plan Ceibal [7]. A propósito de esto, desde este año en el marco de Ceibal se comenzó con un proyecto de enseñanza de Inglés [8]. Éste utiliza la tecnología de videoconferencia para lograr que un docente remoto (ubicado fuera de Uruguay por ejemplo) interactúe con cierta frecuencia con un grupo de alumnos de una escuela pública en particular. Hasta el momento, el proyecto se piensa para las escuelas que cuentan con Fibra Óptica (FO) debido al ancho de banda y retardo necesarios. En un futuro, podría ser extendido a las escuelas que comparten conectividad con otra mediante enlaces inalámbricos punto a punto (ver figura 1.1). Para esto último se necesita lograr que el enlace inalámbrico pueda garantizar ciertos niveles de QoS.

Resumiendo, los proyectos de telemedicina y teleeducación en zonas rurales aisladas se basan mayoritariamente en tecnologías inalámbricas. En estos casos se requiere que estas tecnologías puedan ofrecer calidad de servicio a este tipo de aplicaciones, priorizándolas frente al tráfico no sensible a la calidad de servicio. En la siguiente sección se analizarán las tecnologías inalámbricas más utilizadas en estos proyectos y sus limitaciones para ofrecer QoS.

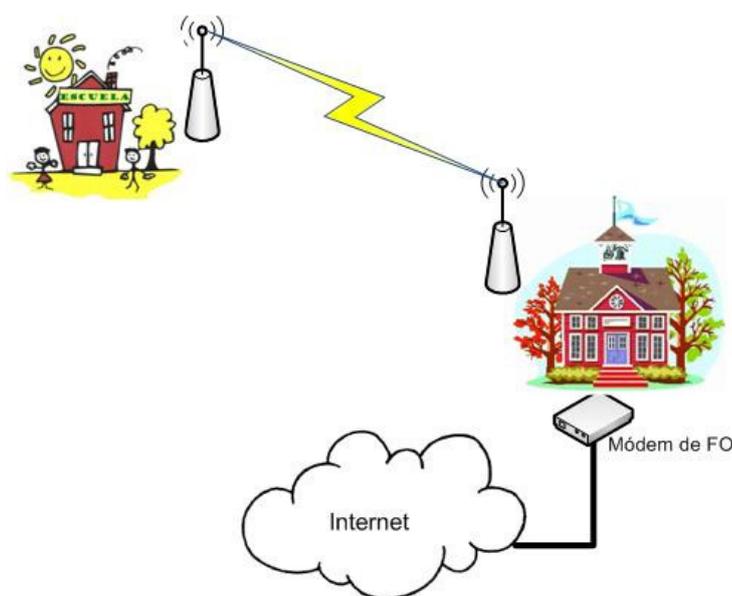


Figura 1.1: Topología de una escuela con conectividad punto a punto que comparte el acceso a Internet con otra.

### 1.1.2. Tecnologías de interés y sus limitaciones

En los últimos años, las Redes Inalámbricas de Área Local (WLANs) y las Redes Inalámbricas de Área Metropolitana (WMANs) se han convertido en alternativas muy populares de acceso a servicios de banda ancha y a Internet. Las principales razones son su bajo costo y su rápido despliegue. Como se mencionó,

## 1.1. Motivación

experiencias previas han demostrado que estas tecnologías (ej.: WiFi) pueden resultar apropiadas para facilitar comunicaciones básicas en zonas aisladas.

La tecnología WiFi (técnicamente definida en las normas IEEE 802.11 [9]) es hoy de uso común para redes locales en oficinas, aeropuertos, centros comerciales, etc. Esta tecnología trabaja en las bandas no licenciadas de 2.4 GHz y 5 GHz. El uso de enlaces WiFi para largas distancias (WiLD) es por tanto un uso anómalo de la tecnología WiFi diseñada para redes de área local con distancias de hasta algunos cientos de metros. Sin embargo, desde hace algunos años se han utilizado con éxito enlaces WiLD en diferentes países [10] [11]. Se han logrado, modificando diferentes aspectos de la capa física (utilizando antenas, amplificadores adecuados y eliminando el multirate) y de la capa de enlace (ajustando diferentes aspectos del protocolo de acceso al medio), enlaces a distancias muy importantes de decenas de kilómetros y en casos experimentales del orden de los cientos de kilómetros [12] [13] [14] [15].

Con la creciente demanda de los usuarios de aplicaciones con requerimientos de tiempo real se planteó un fuerte desafío para las redes 802.11, resultando en el desarrollo de IEEE 802.11e [16]. Esta extensión establece dos modos de operación, uno para QoS parametrizada denominado Hybrid coordination function Controlled Channel Access (HCCA), y otro para QoS priorizada llamado Enhanced Distributed Channel Access (EDCA)<sup>1</sup>. EDCA plantea la posibilidad de brindar distintas prioridades de acceso al medio considerando distintas clases de tráfico. Un mayor detalle de esta función se encuentra en la sección 2.1. Más recientemente se liberó IEEE 802.11n [17] que obliga a incorporar a los fabricantes los mecanismos de acceso al medio definidos en 802.11e.

Es importante resaltar que EDCA no soporta requerimientos estrictos de QoS para aplicaciones de tiempo real como voz o video interactivo, por este motivo, los mecanismos de control de admisión juegan un rol fundamental para poder brindar calidad de servicio en redes 802.11e/n. En 802.11e se prevé la existencia de mecanismos de este tipo pero se deja a los fabricantes la definición de los algoritmos a implementar. Un buen resumen de las principales propuestas sobre este tema se pueden encontrar en los artículos [18] y [19].

Por otro lado, en 2009 el IEEE publicó el estándar 802.16-2009 [20], comúnmente conocido como WiMax: un estándar específicamente diseñado para redes WMAN. Desde el principio WiMax fue considerada como la tecnología para resolver los problemas de comunicación en zonas rurales aisladas, sin embargo, los altos costos y el consumo de los equipos involucrados dificultan su adopción masiva en este contexto. A pesar de ello, esta tecnología ha sido utilizada en zonas rurales para enlaces punto a punto o como parte de una arquitectura híbrida 802.11-802.16

---

<sup>1</sup>Debido a que los dispositivos existentes en el mercado en su gran mayoría implementan sólo EDCA, en este trabajo se decidió analizar y trabajar principalmente con este mecanismo

## Capítulo 1. Presentación

Resumiendo, en las redes rurales se utilizan comúnmente topologías formadas por un enlace punto a punto WiFi (en general de larga distancia) y distribución a los usuarios con enlaces WiFi punto-multipunto. O también se utilizan topologías híbridas que involucran: WiMax para el backbone y WiFi para llegar hasta los clientes.

### 1.1.3. Desafíos

Asegurar la calidad de servicio en una red inalámbrica es un problema complejo y con diversos aspectos para investigar y desarrollar. Sin embargo, toda arquitectura de calidad de servicio requiere de dos herramientas que serán el foco de atención de esta tesis. Por un lado cualquier arquitectura de QoS debe poder analizar el desempeño de la red a los efectos de estimar los parámetros de calidad de servicio que experimentará una nueva conexión de cierto usuario. Por otro lado, como ya se mencionó, una arquitectura de calidad de servicio debe poder controlar el número de aplicaciones de cada tipo, es decir, realizar un control de admisión de las nuevas conexiones, a los efectos de garantizarle a cada clase el servicio adecuado a sus requerimientos.

Para cualquiera de los dos problemas que abordará esta tesis: la estimación de parámetros de QoS de extremo a extremo y el control de admisión, es necesario poder contar con una herramienta que evalúe el desempeño futuro de una nueva conexión. En el primer caso con la estimación del desempeño futuro se podrán tomar decisiones de ingeniería de tráfico, y en el segundo, dependiendo de ciertos umbrales se decidirá si dejarla o no acceder a la red.

En las redes inalámbricas existen varios factores adicionales al análisis del caso cableado que afectan el estado de la red, siendo la probabilidad de colisión y la interferencia externa los considerados más influyentes. El desempeño de extremo a extremo que tendrá una conexión, aún cuando se conozca el estado de la red que ve cada nodo, no es sencillo de estimar directamente, más aún en redes heterogéneas y/o multisalto. Esto se debe a que cada nodo puede “ver” un comportamiento distinto de la red en un momento dado. Un ejemplo de esto es el caso simple de un enlace inalámbrico punto a punto de larga distancia donde un extremo puede estar en una ciudad y el otro extremo en una zona rural prácticamente despoblada. Claramente los niveles de interferencia percibidos por ambos extremos pueden llegar a ser muy distintos. En este caso un nodo podría “pensar” que la red está en buenas condiciones para realizar el envío de cierto tráfico, pero a pesar de esto, el extremo destino no recibe el flujo correctamente debido a la gran interferencia a la que está sometido. Avanzando aún más, si se piensa en brindar conectividad no solamente a un punto sino a varios en una zona de baja densidad de población o si se desea acceder a un sitio donde no exista “línea de vista” con el punto en el que se cuenta con acceso a Internet, es necesario desplegar una red de enrutadores inalámbricos 802.11e en largas distancias o por ejemplo una red híbrida (backbone

## 1.2. Síntesis de la Propuesta

WiMax y acceso WiFi). Ofrecer calidad de servicio de extremo a extremo en estas redes es un tema de investigación actual, aún para el caso de una red local (de cortas distancias) de este tipo.

Lo anterior nos lleva a buscar métodos que analicen el comportamiento de todo el camino globalmente y no de cada nodo individual. Sin embargo, actualmente existen muy pocas herramientas para estudiar la calidad de servicio de extremo a extremo en las redes de interés como se verá en la sección 2.4.

Para el estudio del desempeño de una red en general, existen dos tipos de metodologías básicas. Metodologías basadas en modelos analíticos o basadas en mediciones en la red. También existen métodos que combinan ambas metodologías básicas. Actualmente existen varios trabajos que modelan analíticamente el desempeño de redes inalámbricas, algunos ejemplos significativos son el de Bianchi [21] en 802.11 y Serrano [22] para 802.11e. Muchos de ellos se basan en las mismas hipótesis fuertes para lograr expresiones tratables simplificando el problema. En el trabajo [23] se realiza un estudio de algunas de estas suposiciones concluyendo (verificando experimentalmente y con simulaciones) que muchas de éstas no siempre son válidas, en particular en los entornos y topologías de interés para este trabajo. Más adelante nos referiremos a estas hipótesis con mayor detalle. Otro punto a destacar, es que las expresiones producto de los métodos analíticos, muchas veces son complejas y requieren un alto costo computacional a la hora de usarlas en la práctica.

Por otro lado, existen diversos trabajos que enfocan el problema de estimar parámetros de QoS basándose en mediciones en WLANs o WMANs (ver por ejemplo [24] y sus referencias). Las propuestas basadas en mediciones pasivas en la red son en general aplicables a un nodo o a un enlace con las restricciones mencionadas antes para extraer conclusiones de extremo a extremo a partir de lo que ve cada nodo de la red. Muchas de las propuestas para hacer mediciones de extremo a extremo utilizan por lo tanto mediciones activas sobre la red inalámbrica. En general estas técnicas tienen la principal desventaja en que los procedimientos realizados son intrusivos afectando la QoS de las conexiones existentes en la red.

En este trabajo se eligió un enfoque basado en mediciones activas resultando en una técnica prácticamente independiente de la tecnología inalámbrica de la red en cuestión. Resulta un gran desafío lograr una metodología lo menos invasiva posible dados los recursos escasos de las redes inalámbricas. O sea, se deberá buscar una herramienta de medición que no afecte a los usuarios que están compartiendo dichos recursos.

## 1.2. Síntesis de la Propuesta

En esta tesis se analizó el desarrollo de redes inalámbricas basadas principalmente en IEEE 802.11 y también se incluyó el caso de redes heterogéneas con

## Capítulo 1. Presentación

otras tecnologías como las IEEE 802.16. Por el tipo de escenarios de interés, el foco estuvo puesto en soluciones de bajo costo para la provisión de comunicaciones de tiempo real con calidad de servicio, principalmente en zonas rurales.

Del conjunto de problemas que involucran el estudio de la calidad de servicio en redes inalámbricas, esta tesis se concentró en proponer un sistema de monitoreo que permitiera:

- estimar parámetros de QoS de extremo a extremo
- tomar decisiones de control de admisión

Ambos objetivos bajo la condición de afectar lo menos posible el desempeño de la red al momento de realizar las mediciones.

### 1.3. Principales Resultados

En esta sección se resumirán las contribuciones más importantes de la tesis.

Se desarrolló una metodología basada en mediciones activas en la red y herramientas de aprendizaje estadístico, que permite estimar parámetros de QoS de extremo a extremo en una red inalámbrica. Dicha técnica se probó que puede ser utilizada también para tomar decisiones de control de admisión (aceptar o rechazar nuevas conexiones procurando asegurar garantías de QoS según cada aplicación).

La técnica se basa en entrenar el sistema durante cierto tiempo (usando flujos de aplicación y paquetes de medida) y luego, una vez entrenado y obtenido el modelo, se realiza un monitoreo que consiste en el envío del tren de paquetes de medida. Los paquetes de medida prácticamente no llevan carga útil (salvo estampas de tiempo) y se envían a una velocidad tal que no generan un impacto importante en la red. Se utilizó Support Vector Machines (SVM) como técnica de aprendizaje estadístico supervisado.

Se probó la validez de la técnica tanto en simulaciones como en experimentos de laboratorio con equipamiento comercial. En ambos casos se obtuvieron resultados alentadores. La propuesta realizada es independiente de la tecnología inalámbrica de la red a monitorear. Se verificó en redes 802.11b/g/e y también en el caso de una red híbrida conformada por un tramo WiFi y otro WiMax.

Los primeros resultados de este trabajo fueron publicados en [25]. Recientemente se envió otro trabajo a la revista “International Journal of Network Management” y se está actualmente en espera de la evaluación. Este último recopila todos los resultados obtenidos en la tesis.

## 1.4. Organización del documento

La organización del documento y un breve resumen de cada capítulo se presenta a continuación:

**Capítulo 1. Presentación,** en donde se realiza la presentación del tema indicando la motivación de su realización. Se presenta un resumen de los resultados más importantes del trabajo.

**Capítulo 2. Marco teórico y Estado del Arte,** que comienza describiendo los aspectos más importantes de cada una de las tecnologías involucradas en el estudio. En este capítulo se hace una revisión de los estudios y bibliografía relevante para IEEE 802.11e-EDCA, IEEE 802.16-2009 y Control de Admisión en redes inalámbricas.

**Capítulo 3. Support Vector Machines,** se explican los fundamentos de la herramienta utilizada para realizar el aprendizaje estadístico.

**Capítulo 4. Metodología,** donde se introduce la metodología propuesta explicando los dos casos de aplicación: estimación de parámetros de QoS y decisiones de control de admisión.

**Capítulo 5. Estimador del estado de la red,** en el cual se define el procedimiento para determinar el estimador del estado de la red inalámbrica.

**Capítulo 6. Simulaciones,** se detallan las pruebas realizadas mediante simulaciones en ns-2 considerando distintas topologías de red y protocolos de acceso al medio. Se muestran los resultados obtenidos al aplicar la metodología propuesta.

**Capítulo 7. Validación Experimental,** en este capítulo se describen los escenarios de pruebas que se proponen para validar la técnica de forma experimental, así como la batería de pruebas que se llevó a cabo para verificar el buen funcionamiento. Se incluye una descripción de las características de los equipos, así como del software que se empleó durante el desarrollo del proyecto.

**Capítulo 8. Conclusiones y Trabajo Futuro,** en este último capítulo se ha realizado un resumen de los resultados obtenidos y se proponen futuras líneas de investigación para mejorar y/o extender la metodología aquí presentada.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 2

## Marco Teórico y Estado del Arte

### 2.1. Calidad de Servicio en redes 802.11e

#### 2.1.1. Introducción al estándar 802.11

El estándar IEEE 802.11 [9] aparece en 1997 con velocidades disponibles de 1 Mbps y 2 Mbps. Desde entonces se han creado varios grupos de trabajo para mejorarlo acompañando el avance tecnológico y la demanda de los usuarios. Hoy en día se hablan de 600 Mbps teóricos para el caso de 802.11n [17] (en condiciones óptimas). A continuación se enumeran algunos miembros de la familia 802.11 indicando sus principales características.

Tabla 2.1: Algunas extensiones definidas en IEEE 802.11.

Estándar	Frecuencia	Máx. Rate Físico	Modulaciones y otros
802.11	2,4 GHz	2 Mbps	DSSS/FHSS/IR
802.11a	5 GHz	54 Mbps	OFDM
802.11b	2,4 GHz	11 Mbps	DSSS
802.11g	2,4 GHz	54 Mbps	DSSS/OFDM
802.11n	2,4 GHz / 5 GHz	600 Mbps	MIMO - OFDM

El estándar es muy complejo y existen múltiples referencias donde se explican sus principales características, tanto de capa física, PHY, como de capa de acceso al medio, MAC (ver por ejemplo [26] y sus referencias). Para el desarrollo de la tesis, nos interesa resaltar el mecanismo de acceso al medio propuesto en dicho estándar, por tal motivo a continuación nos centraremos en este tema.

En la capa MAC se especifican dos modos de funcionamiento:

- Point Coordination Function (PCF): Mecanismo centralizado basado en sondeo. La mayoría de los dispositivos del mercado soportan únicamente el modo distribuido dado que resulta más simple su implementación.

## Capítulo 2. Marco Teórico y Estado del Arte

- Distributed Coordination Function (DCF): Mecanismo distribuido en el cual todas las estaciones emplean un protocolo llamado CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance) y un tiempo aleatorio de espera cuando el medio esta ocupado.

En la función DCF básica, cada estación monitorea el canal, si el mismo está libre durante un período de tiempo DIFS<sup>1</sup> intenta transmitir. Sino está libre, persiste monitoreando el canal hasta que se libera. En cuanto se libera, espera un tiempo aleatorio y transmite. Este tiempo aleatorio es el resultado del algoritmo de backoff donde el rango para elegir esta espera aleatoria es llamada Ventana de Contienda (CW), que varía de acuerdo con el número de retransmisiones previas.  $CW$  se inicializa en  $W_{min}$  y se multiplica por 2 por cada retransmisión hasta un valor máximo  $W_{max}$ . El algoritmo de backoff determina el contador de backoff que se traduce en la cantidad de ranuras de tiempo que se tienen que esperar para transmitir. Por cada ranura de tiempo se va decrementando el contador hasta llegar a cero que es el momento en que se transmite. En caso de ocuparse el canal durante la cuenta regresiva, el contador se congela y la estación continúa monitoreando el canal. Una vez que se libera, continúa el proceso para transmitir (se sigue decrementando el contador). Este último comportamiento se observa en la primer estación de la figura 2.1: el contador de backoff llegó a 14 y el medio se ocupó, una vez liberado, continúa el decremento del mismo.

Al transmitir, existen dos posibilidades: que sea exitosa o que no llegue correctamente al destino. El nodo transmisor una vez terminado el envío de la trama, queda escuchando el canal durante un tiempo SIFS<sup>2</sup> en el cual se le debe confirmar la recepción de la misma. Si no se recibe la trama de ACK se programa una retransmisión aumentando la ventana de contienda. En caso de tratarse de una transmisión exitosa  $CW$  vuelve al valor  $W_{min}$ .

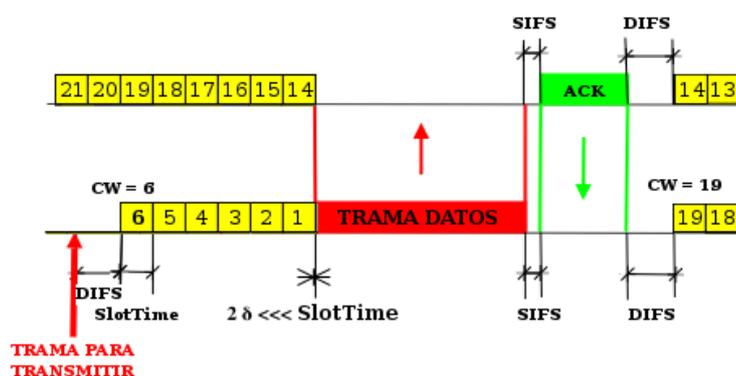


Figura 2.1: Esquema del funcionamiento de DCF (dos estaciones). Fuente [26]

<sup>1</sup>Distributed Inter-frame Space

<sup>2</sup>Simple IFS

## 2.1. Calidad de Servicio en redes 802.11e

El mecanismo de backoff es el responsable de coordinar el acceso al medio compartido tratando de que la probabilidad de colisiones no crezca excesivamente, siempre y cuando las estaciones se escuchen entre si.

Está previsto en la norma un mecanismo de detección virtual de portadora para evitar colisiones cuando se tiene el problema de estación oculta. Se basa en dos mensajes, RTS<sup>3</sup> y CTS<sup>4</sup>, de pedido de envío y respuesta positiva de envío.

### 2.1.2. IEEE 802.11e

Los modos tradicionales de 802.11 (a/b/g) no permiten ningún mecanismo para proveer QoS. Dado el gran desarrollo y despliegue de las redes inalámbricas de los últimos años, sumado a la creciente demanda de aplicaciones multimedia (con estrictos requerimientos de QoS), es que surge 802.11e [16] (aprobado en 2005).

A nivel de capa de enlace, éste propone una nueva función llamada Hybrid Coordination Function (HCF) que define dos modos de operación siguiendo una idea similar a PCF y DCF:

- HCCA: soporta tráfico parametrizado, como se define en IntServ [27]
- EDCA: diseñada para soportar priorización de tráfico, tal como hace DiffServ [28]

Al igual que lo que lo que sucedió con PCF y DCF, el mecanismo EDCA es el elegido a la hora de implementar el hardware correspondiente.

802.11e hace una separación entre las estaciones con soporte de calidad de servicio (QSTA - QoS Enhanced Station) y las que no (STA). Análogamente realiza la misma división entre los Access Points (QAP y AP).

Como se explicó anteriormente, en el estándar original de 802.11 el receptor debe enviar una trama de reconocimiento (ACK) cuando recibe una trama correctamente. Lo anterior deja de ser obligatorio en 802.11e, los ACK de capa MAC pasan a ser opcionales (mejorando el rendimiento en particular cuando se trata de aplicaciones sensibles al retardo).

Se introduce además, un nuevo concepto denominado Oportunidad de Transmisión (TXOP). El TXOP es un intervalo de tiempo acotado en el cual la QSTA está habilitada para transmitir una sucesión de tramas una vez que obtiene el acceso al medio. En DCF, cada vez que una estación ganaba el acceso al medio, le era permitido enviar una única trama. Ahora dependiendo del valor del TXOP se pueden llegar a mejorar los tiempos de transmisión y en algunas situaciones se aumenta throughput total.

---

<sup>3</sup>Request-to-Send

<sup>4</sup>Clear-to-Send

## Capítulo 2. Marco Teórico y Estado del Arte

### EDCA

Utiliza priorización de tráfico en base a cuatro Categorías de Acceso (AC):  $AC\_VO$ ,  $AC\_VI$ ,  $AC\_BK$  y  $AC\_BE$ <sup>5</sup>, donde el comportamiento de cada una de éstas es similar al mencionado para el modo DCF.

Cada AC tiene sus propios parámetros de acceso al medio:

- $CW_{min}$ : valor mínimo de la ventana de contienda.
- $CW_{max}$ : valor máximo de la ventana de contienda.
- $TXOP$ : duración máxima que cada QSTA puede transmitir una vez que gana el acceso al canal.
- $AIFS$  (Arbitrary InterFrame Space): intervalo de tiempo entre que el medio de acceso se coloca en estado idle y el comienzo de una negociación del canal (mismo concepto que el tiempo DIFS explicado para el caso de DCF).

En la figura 2.2 se muestra un esquema que ayuda a comprender las diferencias entre DCF y EDCA.

Configurando apropiadamente estos parámetros es que se consigue la priorización de las distintas clases. A las clases más prioritarias, en general usadas para tráfico en tiempo real, se les setea: mayor  $TXOP$ , menor  $AIFS$  y  $CW_{max}$ , comparado con las de menor prioridad. En los capítulos 6 y 7 se muestran los valores usados para estos parámetros de cada clase en las simulaciones y las pruebas reales respectivamente.

## 2.2. Control de Admisión en redes 802.11e

En la sección anterior vimos que el estándar IEEE 802.11e introduce la posibilidad de brindar ciertos niveles de QoS haciendo priorización de tráfico a la hora de acceder al medio compartido. Para garantizar estrictos requerimientos de QoS (ej.: aplicaciones en tiempo real) y para garantizar un control de congestión en la red, con 802.11e no es suficiente. Es por ello que los mecanismos de control de admisión juegan un rol muy importante a la hora de brindar QoS en redes 802.11e y más general en redes heterogéneas (especialmente cuando se tiene tráfico multimedia).

Un mecanismo de control de admisión representa un componente clave cuando se desea administrar óptimamente los recursos de una red inalámbrica. En otras palabras, la admisión de un nuevo flujo se debe dar si se cumplen las siguientes dos condiciones:

---

<sup>5</sup> $AC\_VO$ : Voz,  $AC\_VI$ : Video,  $AC\_BK$ : Background y  $AC\_BE$ : Best Effort

## 2.2. Control de Admisión en redes 802.11e

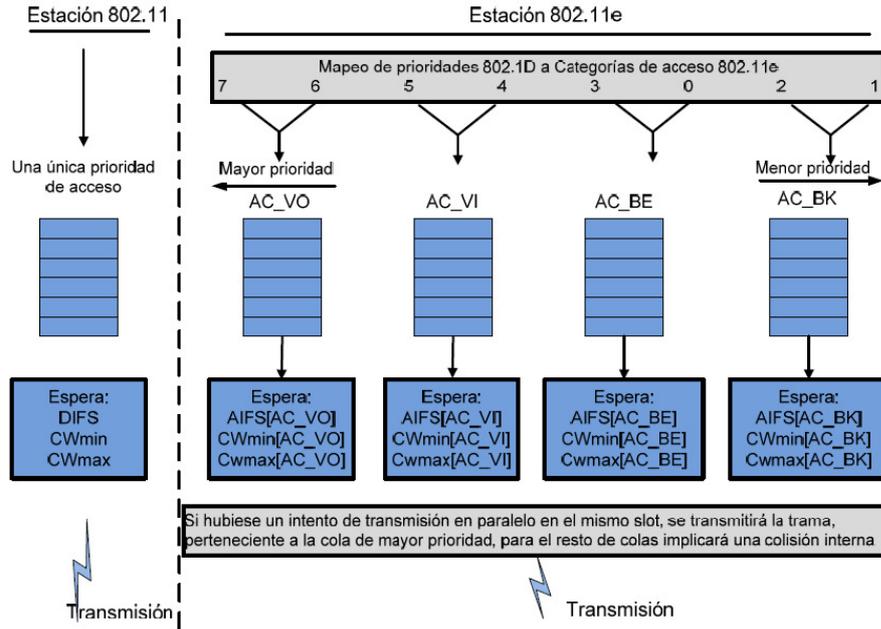


Figura 2.2: Comparación 802.11 (DCF) y 802.11e (EDCA). Fuente [29]

- existen suficientes recursos en la red para garantizar los requerimientos de QoS al nuevo flujo
- la QoS de los flujos existentes en la red no se verá degradada con el ingreso del nuevo flujo

La decisión debe tratar además, de utilizar eficientemente los escasos recursos que se tienen en la red inalámbrica.

El estándar 802.11e deja libre al fabricante la implementación de los algoritmos pero igualmente prevé un cierto intercambio de tramas que permiten realizar un control de admisión a través del QAP. Cuando una QSTA quiere ingresar un nuevo flujo en la red envía primeramente una solicitud de permiso usando una trama ADDTS<sup>6</sup> al QAP. El QAP debe decidir si otorgar el permiso de transmitir o no, para ello se puede fijar en la información que viene en la trama de solicitud y contesta con un ADDTS Response. La decisión la debe realizar de acuerdo a algún algoritmo de control de admisión.

Actualmente existen distintas propuestas para implementar mecanismos de control de admisión tanto para EDCA como para HCCA. Los algoritmos propuestos para EDCA se pueden clasificar en tres clases: basados en mediciones, basados en modelos e híbridos [18] [19].

<sup>6</sup>Add Traffic Stream Request

## Capítulo 2. Marco Teórico y Estado del Arte

Los basados en mediciones (ejemplos: [30] y [31]) basan sus decisiones en monitoreos continuos de la red. Los trabajos actuales en esta área, tienen la gran ventaja de ser simples a la hora de implementar pero la principal desventaja radica a la hora de fijar los umbrales de decisión. Por ejemplo, en [31] lo que se propone es reservar cierto ancho de banda para cada AC (ej.: 20 % para AC\_0, 40 % para AC\_1, etc). Esta partición estática del ancho de banda puede ocasionar una ineficiente utilización de los recursos en determinadas situaciones de carga de la red. Otros algoritmos en vez de usar para decidir el ancho de banda relativo, usan como parámetro de decisión la probabilidad de colisión.

Las propuestas basadas en modelos en general realizan varias suposiciones para lograr obtener expresiones sencillas de manejar pero que en la práctica no siempre son válidas. Tienen además la desventaja de no ser tan flexibles y en general requieren un gran costo computacional. Debido a las suposiciones realizadas, ofrecen distinta precisión en los resultados dependiendo de las condiciones de red, y por lo tanto muchas veces realizan decisiones muy conservadoras que no permiten optimizar los recursos de la red. Un ejemplo de este tipo de trabajos es [32].

Los híbridos tratan de mejorar las debilidades de los anteriores pero todavía no existe una propuesta que sea totalmente aceptada para su implementación.

Los algoritmos revisados en este trabajo están basados particularmente en una única tecnología y son en general aplicables a un enlace. Si a las debilidades mencionadas le sumamos los siguientes escenarios: redes multisalto y redes inalámbricas híbridas (ej.: WiMAX y Wifi), el problema se vuelve más complejo. Por lo anterior, el diseño de mecanismos de control de admisión para redes inalámbricas resulta un tema motivante a encarar.

### 2.3. Calidad de servicio en WiMax

#### 2.3.1. Introducción

Worldwide Interoperability for Microwave Access (WiMax) es el nombre comercial de la tecnología de red de banda ancha inalámbrica que basa su funcionamiento en el estándar IEEE 802.16-2009 [20]. Una de sus ventajas es dar servicios de banda ancha en zonas donde el despliegue de otras tecnologías (ej.: fibra óptica), por la baja densidad de población, presenta unos costos por usuario muy elevados (ej.: zonas rurales).

Un caso de éxito representativo de despliegue WiMax en zonas rurales es el proyecto “Siyakhula Living Labs” [33]. Este fue desarrollado en Sudáfrica por un grupo de investigadores de las universidades Rhodes y Fort Hare. El objetivo fundamental del proyecto era desarrollar y probar en campo una plataforma de comunicación sencilla, rentable y robusta que pueda ser usada en las zonas marginales del país. La motivación principal que llevó a esta iniciativa, es que en Sudáfrica,

## 2.3. Calidad de servicio en WiMax

alrededor del 40 % de la población vive en zonas marginales (o semi- marginales) desprovistos de muchos servicios hoy considerados esenciales. Luego, el objetivo se extendió a atacar las zonas rurales del país. Para lograr la meta propuesta, hasta el momento se tienen interconectadas varias escuelas mediante enlaces WiMax de larga distancia, donde algunos puntos tienen acceso a Internet. Se espera en el correr de 2012 terminar con el despliegue. Se está trabajando para que la plataforma soporte servicios como teleeducación, gobierno electrónico y telemedicina, que permitan bajar los costos y apoyar la actividad económica local.

Una gran ventaja de esta tecnología es que puede operar en bandas no licenciadas. Es importante resaltar que el WiMax Forum [34] aún no ha definido un perfil de certificación para equipos que operan en estas frecuencias, por lo tanto, no sería formalmente correcto hablar de equipos WiMax cuando se trabaja en bandas libres. Aclarado este detalle, en este trabajo se hablará indistintamente de WiMax o del estándar 802.16.

Este estándar fue diseñado para WMAN, en particular se especifica la capa física (PHY) y la capa de control de acceso al medio (MAC). A continuación se resumirán los aspectos que resultan relevantes para la total comprensión de la tesis.

Se define el funcionamiento en una topología celular bajo la modalidad punto-multipunto. En este modo, la arquitectura está conformada principalmente por dos elementos: la estación base (BS) y las estaciones suscriptoras (SS<sup>7</sup>). En una red WiMAX existe una única BS y una o varias SS, donde la BS es la que provee la inteligencia en la red (todas las comunicaciones pasan por la BS y es la encargada de gestionar los recursos).

La comunicación entre las estaciones suscriptoras y la estación base es bidireccional, y la duplexación entre uplink y downlink puede realizarse en frecuencia (Frequency Division Multiplexing o FDD) o en tiempo (Time Division Multiplexing o TDD). Cuando se trabaja en banda libre el método de duplexación especificado en el estándar es TDD.

Para comprender el funcionamiento de WiMAX, en especial lo referente a la capa MAC, el estudio se basó principalmente en el propio estándar. Además se revisaron trabajos previos sobre el tema, ej.: [26], [35] y [36].

### 2.3.2. Calidad de servicio

La capa MAC definida en [20] se caracteriza por estar orientada a conexión y por realizar una planificación determinística para distribuir los recursos de la red entre las conexiones existentes. La provisión de QoS se realiza a través de un

---

<sup>7</sup>También son llamadas Customer Premises Equipment (CPE).

Obs: Las estaciones suscriptoras pueden ser móviles (MS) pero en adelante sólo se hará referencia a las estaciones suscriptoras fijas, que son las de interés en este trabajo.

## Capítulo 2. Marco Teórico y Estado del Arte

mecanismo denominado *Grant/Request*, donde las SS solicitan el ancho de banda que necesitan (*Request*) para cada conexión y la BS asigna los recursos a cada SS (*Grant*) en función del ancho de banda disponible y las características de QoS de cada conexión. Un *Grant* no es más que una asignación de una o más ranuras en la parte de la trama dedicada al enlace ascendente. La BS por su parte, gestiona el enlace descendente, asignando a cada conexión lo que considera (no tiene que interactuar con otra estación para decidir esta asignación).

Cada conexión debe estar asociada a un flujo de servicio (relación 1 a 1). Los flujos de servicio están definidos en la BS y contienen un conjunto de parámetros con las características de QoS para un cierto tipo de tráfico (varios flujos de aplicación pueden ir asociados a un mismo flujo de servicio). Los flujos de servicio son unidireccionales (uplink o downlink).

Cada flujo de servicio tiene asociado, en base a sus requerimientos de QoS, un servicio de planificación.

La MAC WiMax soporta cinco tipos distintos de servicios de planificación:

- Unsolicited Grant Service (UGS): diseñado para flujos en tiempo real con estrictos requerimientos de retardo, con paquetes de datos de tamaño fijo y transmitidos en intervalos periódicos de tiempo. Una vez asignado un servicio UGS a una SS, la BS asigna periódicamente el ancho de banda requerido, sin necesidad de una previa solicitud. Con esto se evita la sobrecarga y el retardo que introducen las solicitudes de las estaciones. Puede emplearse por ejemplo para VoIP sin supresión de silencios. El conjunto de los parámetros de QoS que un flujo de servicio UGS debe definir son:
  - Minimum Reserved Traffic Rate
  - Maximum Sustained Traffic Rate
  - Maximum Latency
  - Tolerated Jitter
- Real-Time Polling Service (rtPS): Diseñado para soportar flujos de datos en tiempo real (con requerimientos menos exigentes de retardo que el caso anterior) que generan paquetes de longitud variable transmitidos en forma periódica (ej.: video codificado en formato MPEG<sup>8</sup> o VoIP con supresión de silencios). En este caso la SS tiene reservada una ranura para solicitar ancho de banda periódicamente, especificando el tamaño deseado para el intervalo de transmisión. Los parámetros a especificar para el flujo de servicio correspondiente son:
  - Minimum Reserved Traffic Rate
  - Maximum Sustained Traffic Rate

---

<sup>8</sup>Moving Pictures Experts Group

## 2.4. Antecedentes de trabajos relacionados con la estimación de parámetros de QoS de extremo a extremo en redes inalámbricas

- Maximum Latency

Éste aumenta la sobrecarga, si se lo compara con UGS, pero aumenta la eficiencia en el transporte de datos al permitir que los intervalos de transmisión tengan tamaño variable.

- Extended Real-Time Polling Service (ertPS): Es un servicio de planificación que se basa en las ventajas de UGS y rtPS. En este caso, la BS puede asignar ancho de banda a una SS sin previa solicitud como ocurre en UGS, pero a diferencia de éste, en ertPS la asignación puede ser de tamaño variable (con previa solicitud de la estación) mientras que en UGS era de tamaño fijo. Este servicio está diseñado para soportar flujos de servicio en tiempo real que generen paquetes de tamaño variable en forma periódica, tales como VoIP con supresión de silencios. Es opcional por lo que no todos los sistemas lo implementan.
- non Real-Time Polling Service (nrtPS): Diseñado para flujos de servicio que no son en tiempo real, que generan ráfagas de paquetes de datos de tamaño variable, como por ejemplo FTP (File Transport Protocol). Lo habitual es que los servicios transportados en éstas conexiones toleren mayor retardo y sean prácticamente insensibles a la variación del mismo. Parámetros principales a configurar:
  - Minimum Reserved Traffic Rate
  - Maximum Sustained Traffic Rate
- Best Effort (BE): Diseñado para flujos de datos que no requieren un nivel de servicio mínimo y que pueden ser transmitidos cuando existe ancho de banda disponible (ej.: navegación por Internet). Los flujos BE deberán solicitar intervalos de transmisión y su solicitud solo será atendida si quedan slots libres después de dar servicio a los flujos con mayor prioridad. En este caso sólo se configura el parámetro: Maximum Sustained Traffic Rate.

## 2.4. Antecedentes de trabajos relacionados con la estimación de parámetros de QoS de extremo a extremo en redes inalámbricas

### 2.4.1. Introducción

La estimación de parámetros de QoS de extremo a extremo tiene gran importancia en especial en las redes inalámbricas. Conociendo las pérdidas, el throughput, etc, es posible poder realizar algún tipo de ingeniería de tráfico.

Dada la naturaleza de las redes inalámbricas existe una gran cantidad de fenómenos que pueden afectar la calidad percibida por los usuarios. Ejemplos de

## Capítulo 2. Marco Teórico y Estado del Arte

éstos son: fading, presencia de obstáculos, interferencias externas, adaptación del rate físico, estación oculta, etc.

Realizar un modelo que tenga en cuenta todos estos fenómenos resulta un desafío muy importante. Existen dos grandes alternativas a la hora de modelar el comportamiento de este tipo de redes, una es trabajar en base a modelos analíticos y otra opción es en base a mediciones. Podríamos considerar también una alternativa híbrida que consiste en medidas en la red conjuntamente con modelos analíticos.

A continuación se describen algunos de los trabajos existentes hasta el momento que permiten la estimación de distintos parámetros de QoS en redes inalámbricas. La mayoría de los estudios presentados para redes inalámbricas están basados en 802.11 por ser la tecnología más popular.

### 2.4.2. Revisión de algunas propuestas en el área de modelos analíticos

Tal como se explicó en la introducción, actualmente existen varios trabajos que modelan el desempeño de una red 802.11 [22] [37] [38] [32]. La mayoría de las propuestas que realizan un modelo analítico están basadas en el paper de Bianchi [21]. En este último se presenta una cadena de Markov que permite determinar distintos parámetros de interés en una red 802.11 como lo son: probabilidad de colisiones, throughput de la celda, etc. Éstos se obtienen en función de los parámetros que intervienen en el mecanismo de acceso al medio (CW, DIFS, etc).

Todos estos modelos en general consideran un canal ideal en donde los errores sólo se deben a las colisiones (no hay estación oculta, ni se considera interferencia). En el trabajo de Duffy [23] se realiza un estudio muy completo de las hipótesis comunes utilizadas en estos trabajos. A modo de ejemplo, a continuación se citan dos de las hipótesis analizadas:

- En una red 802.11, la secuencia de eventos generados por el éxito o fracaso en los sucesivos intentos de transmisión de una cierta estación forman una secuencia de eventos aleatorios independientes y equidistribuidos, y en particular que no dependen de la historia de colisiones
- En los modelos que se consideran estaciones con buffer no nulo, la secuencia de variables que representan el estado de la cola (si tiene paquetes o si está vacía) está formada por variables aleatorias independientes idénticamente distribuidas y en particular no dependen del estado de backoff.

Se verifica, mediante simulaciones y de forma experimental que estas hipótesis no siempre son válidas. Dependen mucho de si el tráfico está en condiciones de saturación o no, dependen del tamaño de los buffer, etc.

Otro punto importante a resaltar es que la mayoría de los modelos analíticos son muy dependientes de la tecnología de la red inalámbrica y son en general para

## 2.4. Antecedentes de trabajos relacionados con la estimación de parámetros de QoS de extremo a extremo en redes inalámbricas

un único salto (punto a punto ó punto a multipunto), ej.: 802.11 sin adaptación de rate [21], 802.11 con adaptación de rate [39], 802.11e [22], WiMax [40], etc. Esto hace difícil su aplicación en redes híbridas o multisalto.

### 2.4.3. Revisión de algunas propuestas en el área de mediciones

En una red inalámbrica, una técnica basada en medidas debe ser lo menos intrusiva posible y debe tener un rápido tiempo de convergencia, más aún en los casos que se desea establecer un monitoreo continuo. Las mediciones pueden ser pasivas o activas, en las pasivas no se inyecta tráfico de medida en la red, pero suelen ser menos precisas en comparación con las activas.

Existen variadas propuestas a la hora de estimar el ancho de banda disponible de extremo a extremo. Para las redes cableadas este tema está muy desarrollado, PathLoad [41] [42], PathChirp [43], y Spruce [44] son ejemplos interesantes. Todas estas técnicas están basadas en medidas activas. En redes cableadas, éstas obtienen buenas precisiones, pero en su mayoría resultan intrusivas y tienen tiempos de convergencia largos. Debido a lo mencionado anteriormente, estas técnicas no son adecuadas para ser utilizadas en redes inalámbricas. Además, se ha comprobado que, dada la variabilidad que se da en la capacidad de un enlace inalámbrico, éstas obtienen estimaciones inexactas.

Específicamente en redes inalámbricas, existen algunas propuestas para la estimación de ancho de banda disponible. En general son muy dependientes de la tecnología de la red y/o de otras características que las hacen poco escalables. Por ejemplo, hay técnicas basadas en redes 802.11 que asumen que el mecanismo RTS/CTS está siempre habilitado (ejemplos: [45] [46]). Otras técnicas resultan muy intrusivas porque las medidas implican saturación de la red.

En el trabajo [47] se define una técnica denominada WBest. Ésta, también para redes 802.11, permite estimar el ancho de banda disponible en una red con la condición de que el último salto sea el enlace inalámbrico menos comprometido (esto hace que no sea muy escalable porque no siempre se da esta situación). En el trabajo mencionado se realiza además una comparación con otras técnicas existentes resultando WBest la menos intrusiva, con muy buena precisión y con costos de convergencia bajos.

Es importante resaltar que la medida de la capacidad media disponible (resultado de las técnicas mencionadas) no es fácilmente utilizable para estimar la QoS que pueda sufrir una aplicación específica, más aún con las variaciones que se presentan en las redes inalámbricas.

No se encontraron trabajos basados en medidas que puedan predecir el desempeño (parámetros de QoS) que tendría cierta aplicación sin dejarla entrar previamente en la red.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 3

## Support Vector Machines

### 3.1. Introducción

A la técnica que consiste en aprender modelos generales a partir de datos tomados de ejemplos particulares se la conoce con el nombre de Aprendizaje Estadístico. Existen sistemas basados en aprendizaje supervisado y no supervisado. Estos últimos tienen como objetivo principal agrupar o organizar los datos de entrenamiento (éstos no son los utilizados en la tesis).

Por otro lado, la idea básica de los sistemas de aprendizaje supervisado es: se tiene un conjunto de muestras de entrenamiento  $(x, f(x))$  para alguna función desconocida  $f$  y el objetivo es encontrar una buena aproximación para dicha función. Posteriormente al entrenamiento se aplica una fase de prueba o verificación, en la cual se analiza la calidad del modelo generado.

En el presente trabajo se consideraron dos problemas de aprendizaje: regresión y clasificación. La regresión se va a usar para el caso de predicciones de los parámetros de QoS y la clasificación para el caso de tomar decisiones de control de admisión.

Existen varias técnicas de aprendizaje estadístico, Support Vector Machines (SVM) es una de ellas. Esta fue introducida por Vapnik en los 90 [48] y desde entonces ha sido muy utilizada y desarrollada mostrando un alto desempeño tanto a la hora de abordar problemas de clasificación como problemas de regresión. Algunos trabajos donde se ha utilizado SVM obteniendo muy buenos resultados son: [49] [50] [51]. En particular, en el trabajo [50] se realiza una comparación de SVM respecto a otra técnica de regresión, obteniéndose mejores resultados para el caso de SVM (requiere menor costo computacional lo que resulta importante si se desean realizar estimaciones en línea como es el caso de la presente tesis).

Dado que la comparación de las distintas técnicas no es el tema central de este trabajo y por los argumentos antes mencionados, es que se eligió utilizar SVM co-

## Capítulo 3. Support Vector Machines

mo método de aprendizaje estadístico. Una descripción completa de la herramienta se encuentra en [52] y [53].

A continuación, en primer lugar se estudiarán las ideas generales de SVM para el caso de regresión y luego se extenderá la explicación para el caso de clasificación.

### 3.2. Regresión con SVM (SVR)

Suponemos que se tiene un conjunto de datos  $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset \Psi \times \mathbb{R}$ , donde  $\Psi$  representa el espacio de estados de las entradas (ej.:  $\Psi = \mathbb{R}^d$ ). El objetivo es estimar la función  $f(x)$ , siendo  $y = f(x)$ .

Para simplificar la explicación, se considerará primero el caso lineal:

$$\hat{f}(x) = \langle x, \tilde{\beta} \rangle + \beta_0$$

donde  $x \in \mathbb{R}^d$ ,  $\beta_0 \in \mathbb{R}$  y  $\langle \cdot, \cdot \rangle$  representa el producto escalar en  $\mathbb{R}^d$ .

En este caso, SVM tiene como objetivo encontrar el hiperplano óptimo  $\langle x, \tilde{\beta} \rangle + \beta_0$  que cumpla que la distancia de todos los puntos  $(x_i, y_i)$  al hiperplano sea menor o igual a  $\varepsilon$  para  $i = 1, \dots, l$ . El problema de regresión puede escribirse de la siguiente forma:

$$\begin{aligned} & \underset{\tilde{\beta}}{\text{mín}} \frac{1}{2} \|\tilde{\beta}\|^2 \\ & \text{sujeto a:} \\ & y_i - \langle x_i, \tilde{\beta} \rangle - \beta_0 \leq \varepsilon, \quad \forall i = 1, \dots, l \\ & \langle x_i, \tilde{\beta} \rangle + \beta_0 - y_i \leq \varepsilon, \quad \forall i = 1, \dots, l \end{aligned}$$

Un supuesto clave en el problema de optimización anterior es que exista solución factible. Obviamente, esto no es siempre el caso, por lo tanto se debe permitir la posibilidad de que algunos puntos no pertenezcan al “tubo de radio  $\varepsilon$ ” (ver figura 3.1). Entonces, el problema puede ser reformulado de la siguiente manera (modificando las restricciones y penalizando con un costo los puntos que quedan fuera):

$$\begin{aligned} & \underset{\tilde{\beta}}{\text{mín}} \frac{1}{2} \|\tilde{\beta}\|^2 + C \sum_{i=1}^l (\zeta_i + \zeta_i^*) \\ & \text{sujeto a:} \\ & y_i - \langle x_i, \tilde{\beta} \rangle - \beta_0 \leq \varepsilon + \zeta_i, \quad \forall i = 1, \dots, l \\ & \langle x_i, \tilde{\beta} \rangle + \beta_0 - y_i \leq \varepsilon + \zeta_i^*, \quad \forall i = 1, \dots, l \\ & \zeta_i, \zeta_i^* \geq 0 \end{aligned} \tag{3.1}$$

La constante  $C > 0$  determina cuanto se penaliza a los errores de entrenamiento (o sea, la cantidad tolerada de desviaciones de más de  $\varepsilon$ ).

### 3.2. Regresión con SVM (SVR)

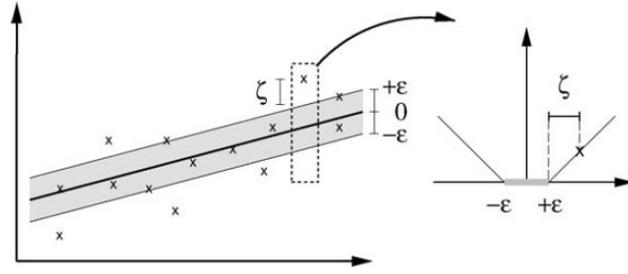


Figura 3.1: SVR: Margen de ajuste (Fuente: [53])

El problema puede ser tratado mediante su problema dual utilizando la formulación de *Lagrange*. Usando las condiciones de *Karush-Kuhn-Trucker* (KKT [54]) se obtiene:

$$\hat{f}(x) = \sum_{i=1}^l (\alpha_i^0 - \alpha_i^{0*}) \langle x_i, x \rangle + \beta_0^* \quad (3.2)$$

donde  $\alpha_i^0$ ,  $\alpha_i^{0*}$  son los multiplicadores de Lagrange correspondiente a los dos conjuntos de restricciones del problema.  $\beta_0^*$  se obtiene de las condiciones KKT y sólo depende del producto interno.

De las condiciones de KKT se desprende que gran parte de los  $\alpha_i^0, \alpha_i^{0*}$  son cero, en particular éstos son los correspondientes a los puntos que quedan dentro del “tubo de radio  $\epsilon$ ”. Sólo los puntos tales donde  $\alpha_i^0, \alpha_i^{0*} \neq 0$  son los necesarios para estimar  $f$ . A dichos puntos se los llama “support vectors” (SV) que dan el nombre a la técnica SVM.

Si se observa la ecuación 3.2 el problema de regresión depende sólo del producto interno entre los datos. En este sentido, para aplicar SVR en un espacio general  $\mathcal{F}$  es sólo necesario conocer el producto interno en dicho espacio. Esta información resulta relevante para el caso no-lineal.

Si no existe un ajuste lineal entre los datos, la idea es mapear los vectores de  $\mathbb{R}^d$  a un espacio de dimensión mayor donde sí se encuentre una buena aproximación para la función  $f$ . Utilizando lo anterior, la idea en el caso no-lineal es transformar los datos de entrada en un espacio de dimensión mayor que denominaremos  $\mathcal{F}$ . El mapeo nos lleva a la siguiente solución:

$$\hat{f}(x) = \sum_{i=1}^{N_s} (\alpha_i^0 - \alpha_i^{0*}) K(x_i, x) + \beta_0^*$$

donde  $N_s$  es el número de support vectors y  $K(x_i, x)$  (denominada kernel) es la función transformación que permite obtener el producto interno de dos vectores del espacio  $\mathcal{F}$ . Alcanza sólo con conocer la función Kernel y no la transformación en si (tampoco es necesario conocer el espacio  $\mathcal{F}$ ).

### 3.3. Clasificación con SVM (SVM-C)

Una vez que la técnica fue explicada para el caso de regresión, el problema de clasificación resulta más simple de analizar. Para simplificar, se mencionará el problema de separación en dos clases (clasificación binaria).

Cuando se trata de una clasificación binaria, que es el caso de uso de este trabajo al momento de implementar decisiones de control de admisión, se tiene el conjunto de datos de entrada de la siguiente forma:  $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset \Psi \times \{+1, -1\}$  donde  $\Psi$  es el conjunto de patrones o muestras de entrada.

En este caso el problema de optimización se reduce a encontrar el hiperplano que maximice el margen de separación entre las  $l$  muestras de entrenamiento, en otras palabras, se desea encontrar el hiperplano que mejor separe los datos. Para encontrar este hiperplano se sigue un razonamiento similar al mencionado para el caso de regresión (usando el Lagrangiano y las condiciones KKT). Una vez encontrado, dada una muestra  $x_j \in \Psi$ , se analizará de que lado del hiperplano queda ubicada y en base a esta información se dirá a que clase pertenece ( $y_j$ ).

La regla de decisión será  $signo(\langle x, \tilde{\beta} \rangle + \beta_0)$ . Una vez realizado todo el análisis se llega a la función buscada:

$$\hat{f}(x) = signo\left(\sum_{i=1}^l \alpha_i^0 y_i \langle x_i, x \rangle + \beta_0^*\right)$$

donde los vectores de soporte son aquellos puntos que están a distancia mínima del hiperplano.

En el caso no lineal, la idea es la misma que en el caso de regresión. Se mapean los datos a un espacio de dimensión mayor y para estimar el hiperplano buscado se utilizan las funciones Kernel.

En las figuras 3.2 y 3.3 se muestran dos casos, uno donde las muestras son linealmente separables y otro donde se observa que la separación de las muestras puede ser más sencilla en un espacio de mayor dimensión.

### 3.4. Implementación de SVM elegida

Existen varias implementaciones de SVM, en particular se utilizó LIBSVMv2.91 – A library for Support Vector Machines [55]. Los kernels más usados son:

- Polinómicos de grado  $d$  de la forma:

$$k(x, x') = (x \cdot x')^d \tag{3.3}$$

### 3.4. Implementación de SVM elegida

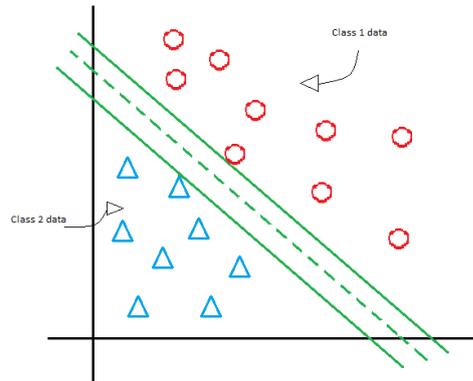


Figura 3.2: SVM-C: Separables linealmente. La línea punteada representa el hiperplano buscado. Las líneas enteras representan los márgenes determinados por los SV.

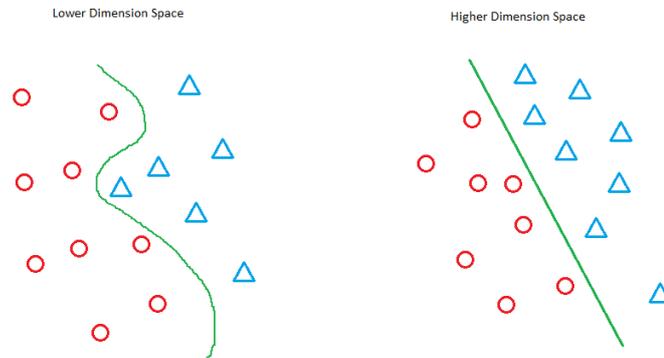


Figura 3.3: SVM-C: Caso no linealmente separable. En este caso se mapean las muestras a un espacio de dimensión mayor donde sí son separables mediante un hiperplano.

- de base radial gaussiana con parámetro  $\gamma$ :

$$K(x, x') = e^{-\gamma \|x - x'\|^2} \quad (3.4)$$

- sigmoide de parámetros  $\kappa$  y  $\theta$ :

$$K(x, x') = \tanh(\kappa(x \cdot x') + \theta) \quad (3.5)$$

En particular, en este trabajo se utilizó el kernel de base radial gaussiana que es comúnmente usado los trabajos mencionados.

A partir de las trazas de las simulaciones o de los logs en el caso de las pruebas experimentales se construye un conjunto de muestras de entrenamiento y otro conjunto de verificación. El objetivo es determinar los valores de  $C$  y  $\gamma$  que maximicen la precisión del modelo, siendo  $\gamma$  la constante en el kernel elegido y  $C$  el costo de penalización al realizar los entrenamientos (definido en las ecuaciones anteriores).

### Capítulo 3. Support Vector Machines

Para ello, los autores proponen en [56], la generación de diferentes modelos recorriendo una grilla de valores de  $C$  y  $\gamma$ . Para recorrer esta grilla, se realizó un script que prueba distintos modelos variando ambos parámetros a partir de incrementos grandes de forma lineal (ej.: 0.1, 1, 10, 100...). Una vez encontrada la zona en la que se sospecha que puede estar el máximo, se realiza una iteración con un salto más pequeño logrando una mejor resolución, encontrando así la combinación  $(C, \gamma)$  con la que se obtiene el mejor resultado.

La implementación de SVM elegida consta de dos ejecutables básicos: “svm-train” y “svm-predict”. El primero usado para generar el modelo partiendo de la muestra de entrenamiento y el segundo se utiliza para verificar la precisión de dicho modelo (a partir del conjunto de verificación).

Parte II

DETALLE DE LA PROPUESTA



# Capítulo 4

## Metodología

### 4.1. Introducción

Como se mencionó anteriormente, modelar las redes inalámbricas de forma analítica resulta muy complejo más aún si se desean evitar ciertas hipótesis, que como vimos no siempre son ciertas. Más difícil resulta el estudio analítico cuando se trata de redes constituidas por tramos de distintas tecnologías.

En este trabajo se toma un enfoque de “caja negra” para analizar el desempeño de una red inalámbrica. Se propone una metodología que consiste en una combinación de mediciones activas en la red conjuntamente con la aplicación de técnicas de aprendizaje estadístico.

A la hora de estimar parámetros de QoS, si lo que se quiere es reflejar el desempeño de cierta aplicación dado un cierto estado de la red, la forma más fácil es inyectar un tráfico similar a la misma (ej.: enviar un video o realizar una descarga de archivos) y a partir de esto, medir el parámetro de interés. Sin embargo, el volumen de tráfico generado de esta manera podría sobrecargar la red afectando la calidad percibida por los usuarios presentes en ese momento. En el otro extremo, se podría optar por mandar sólo paquetes pequeños que no sobrecarguen la red y a partir de su desempeño extrapolar al que tendría la aplicación deseada (esto claramente no siempre es cierto, depende mucho de la aplicación).

Como solución intermedia se presenta una metodología que se basa en la utilización de herramientas de aprendizaje estadístico, donde el sistema es entrenado durante ciertos períodos con flujos de aplicación y con la inyección de una ráfaga de paquetes de medida. Una vez entrenado el sistema, sólo es necesario enviar el tren de paquetes de medida y analizando la estadística de las variaciones en sus tiempos entre arribos se puede estimar el parámetro de desempeño deseado para cierta aplicación sin sobrecargar significativamente la red.

Una técnica similar es aplicada en [50] pero para el caso de redes cableadas

## Capítulo 4. Metodología

donde la situación es bastante distinta. En redes inalámbricas se tienen otros problemas como lo son por ejemplo el de la estación oculta, desvanecimiento, interferencia, etc. Pero al igual que en una red cableada, se tiene un ancho de banda compartido y los distintos parámetros de QoS están fuertemente ligados a la cantidad de conexiones presentes en la red.

La tesis de este trabajo parte de que dado el comportamiento de los paquetes de medida es posible obtener una estimación del estado de la red de extremo a extremo (considerando el camino que cursan dichos paquetes de medida). Por ejemplo, si la red presenta bajas colisiones y bajos niveles de interferencia externa, se espera que los tiempos entre arribos de los paquetes de medida sean muy parecidos a sus tiempos entre partidas. Sin embargo, en un escenario donde hay muchas colisiones, los tiempos entre arribos pueden sufrir grandes modificaciones. Una ilustración de lo mencionado se puede ver en la figura 4.1.

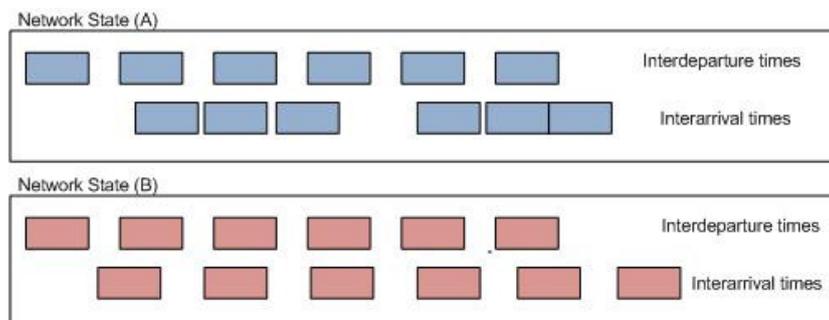


Figura 4.1: Tiempos entre arribos de los paquetes de medida para dos estados diferentes de la red. A: Muchas colisiones y/o alta interferencia. B: Bajas colisiones y baja interferencia.

Existe un compromiso entre la cantidad de paquetes de medida y su tamaño a la hora de lograr buenas estimaciones, con buenos tiempos de convergencia y tratando de ser lo menos intrusivo posible. Se elige trabajar con paquetes pequeños (casi sin carga útil) que son enviados a cierta velocidad buscando lograr el menor impacto posible en la red y rápidos tiempos de convergencia. Esto último se explica, ya que al aumentar el tamaño de los paquetes de medida, para lograr un buen número de muestras para entrenar y/o monitorear, debe aumentar el tiempo de monitoreo y por lo tanto aumenta el tiempo de convergencia. Con paquetes pequeños se logra el efecto contrario, además de sobrecargar menos la red.

Para simplificar el análisis, se considerará que los paquetes de medida se envían con un tiempo entre-paquetes constante y son de tamaño fijo (con la menor cantidad de bytes posibles en el payload). Para dichos paquetes se usará UDP como protocolo de capa de transporte ya que simplemente se requiere el envío y no son necesarias otras funcionalidades que permite TCP como lo son: el acuse de recibo, control de congestión, etc.

## 4.2. Estimación de parámetros de QoS

En las siguientes dos secciones se detallará la aplicación de la metodología para los dos problemas mencionados anteriormente: estimación de parámetros de QoS y Control de Admisión.

### 4.2. Estimación de parámetros de QoS

En este caso, el uso de la metodología es con el objetivo de aprender la relación entre el comportamiento de los tiempos entre arribos de los paquetes de medida y los parámetros de QoS que experimenta cierto flujo de tráfico correspondiente a alguna aplicación. Una vez conocida esta relación, se podrá estimar el parámetro de QoS deseado sólo enviando el tren de paquetes de medida.

Se parte del modelo de regresión  $Y = \Phi(X) + \epsilon$ , donde  $X$ ,  $Y$  y  $\epsilon$  son variables aleatorias.  $Y$  representa el parámetro de QoS a estimar,  $X$  es el estimador del estado de la red y  $\epsilon$  los errores (variable aleatoria con media nula e independiente de  $X$ ).

Para explicar la metodología haciendo referencia a los casos de verificación, se define un escenario que consiste en una red inalámbrica formada por un nodo de referencia (RN) y varios nodos fijos<sup>1</sup> distribuidos aleatoriamente en cierta superficie (ilustrado en la figura 4.2). En este caso se considerará 802.11 “puro” (sin calidad de servicio) donde todos los nodos usan el mecanismo DCF para acceder al medio (se podría generalizar a una red inalámbrica cualquiera). Se denominará Nodox al cliente al que se le quiere estimar los parámetros de QoS (ej.: throughput, delay, jitter, etc) que percibiría si entrara en la red y estableciera una conexión con el RN. Llamaremos al Nodox “cliente nuevo” y a la conexión entre él y RN “nueva conexión”.

El objetivo se traduce en estimar la función  $\Phi$ , esto se realizará a partir de observaciones de las variables  $(X, Y)$ . Una vez estimada  $\Phi$  se puede validar el modelo o se pueden hacer predicciones sobre la variable  $Y$ . Para esto, se divide el algoritmo en dos fases: Fase de Entrenamiento y Fase de Monitoreo.

#### Fase de Entrenamiento

Para entrenar al sistema, el Nodox inyecta tráfico de aplicación en la red con destino el RN y se registra el throughput<sup>2</sup> ( $Y_i$ ) obtenido por dicha conexión. Cuando culmina la transmisión anterior, el Nodox envía el tren de paquetes al nodo RN (los paquetes de medida deben experimentar el mismo camino inalámbrico que el flujo de aplicación anterior). Considerando los tiempos de partida y arribo de dichos paquetes se construye  $X_i$  (el estimador del estado de la red) que más adelante se definirá en detalle.

---

<sup>1</sup>Una vez sorteadas las posiciones los nodos no se mueven.

<sup>2</sup>Podría tratarse de otro parámetro de QoS

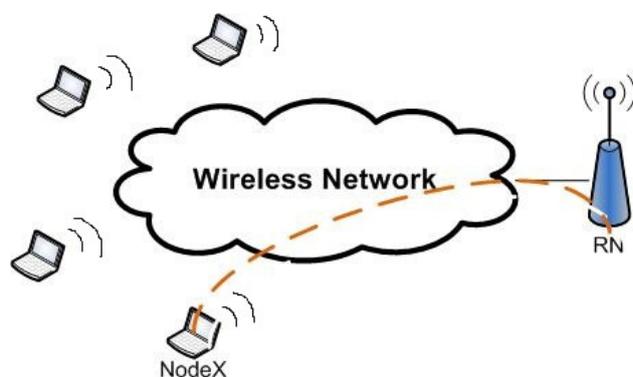


Figura 4.2: Escenario usado para explicar la metodología.

Para lograr un buen modelo, durante el entrenamiento se deben realizar varios experimentos variando las características del estado de la red (aumentando la cantidad de conexiones, ubicando los nodos en distintas posiciones, etc). En cada experimento se obtiene una pareja  $(X_i, Y_i)$ . Una vez recolectada la suficiente cantidad de muestras se estima la función  $\Phi$  usando SVM (en particular SVR). Denominaremos  $\hat{\Phi}$  al estimador de la función  $\Phi$ . Una vez que el sistema está entrenado comienza a correr la segunda fase.

La fase de entrenamiento sólo debería ejecutarse en los casos que hay un cambio drástico en la red. En topologías relativamente estáticas, es de esperar que sea necesario sólo entrenar una vez desplegada la red y luego no se necesite hacerlo con asiduidad (salvo que exista algún cambio en algún equipo o de configuración por ejemplo).

## Fase de Monitoreo

En esta fase, sólo se envía la ráfaga de paquetes de medida desde NodoX al RN y se construye  $X$  de la misma forma que en la fase anterior. El throughput que obtendría una conexión entre los nodos mencionados ( $\hat{Y}$ ) se puede estimar de la siguiente forma:  $\hat{Y} = \hat{\Phi}(X)$ .

Es interesante destacar que la fase de monitoreo (que se ejecuta cada vez que se desea estimar cierto parámetro de QoS) no sobrecarga la red. En el correr del documento se seguirá trabajando con la estimación del throughput, pero es importante aclarar que la metodología podría ser utilizada para predecir otros parámetros de QoS como por ejemplo: jitter, delay, probabilidad de pérdidas, etc.

## 4.3. Herramientas para Decisiones de Control de Admisión

La operación básica de un mecanismo de control de admisión es decidir cuándo una conexión debe ser admitida en la red y cuando no. En esta sección se discu-

### 4.3. Herramientas para Decisiones de Control de Admisión

tirá como utilizar la metodología presentada anteriormente para dar los primeros pasos a la construcción de un algoritmo de este estilo. O sea, la idea es obtener un mecanismo de clasificación que se base en el estado de la red y requerimientos de QoS de cierta aplicación particular para decidir cuándo aceptar una nueva petición de conexión.

Para explicar la metodología para este caso de uso, se considera el escenario de una red en la que pueden cursarse distintas clases de tráfico con distintas prioridades. Como se explicó y como se verificará en el capítulo 7, la técnica resulta independiente de la tecnología inalámbrica en cuestión.

En este caso se considera la misma topología detallada en a Figura 4.2, pero con la diferencia de que se trata de una red en la que los nodos son estaciones que soportan QoS (por ejemplo EDCA como método de acceso al medio si se tratara de una red 802.11e). Para simplificar el problema se considerará que existen únicamente dos clases de tráfico, por ejemplo: VoIP (clase 1) y Best Effort (clase 2). Toda la explicación siguiente es válida cuando tengo sólo una clase de servicio (que sería el caso de 802.11b/g “puro” por ejemplo).

Se considerará un criterio de clasificación del tipo ON-OFF (clasificación binaria) dependiendo de cierto umbral definido para cada clase ( $maxDelay$  y  $minTh$  respectivamente). Esto significa:

- Un nuevo flujo de clase 1 será aceptado en la red si se estima que su delay será menor al umbral  $maxDelay$ , caso contrario será rechazado.
- Un nuevo flujo de clase 2 será aceptado en la red si se estima que su throughput superará el mínimo permitido ( $minTh$ ), sino será rechazado.

*Observación:* Se eligieron delay y throughput, pero se podrían haber elegido otros parámetros (ej.: jitter o probabilidad de pérdida de paquetes). Para simplificar se tomó un único parámetro de decisión por clase.

El algoritmo se divide en las mismas dos fases mencionadas anteriormente: entrenamiento y monitoreo. Durante el entrenamiento se obtiene un conjunto de tuplas  $(X_i, Z_i)$ , donde  $X_i$  se construye de la misma forma que el caso anterior.  $Z_i$  representa una secuencia binaria que resulta de comparar el parámetro de clasificación (throughput o delay) con su umbral correspondiente ( $minTh$  o  $maxDelay$ ).

Utilizando SVM-C se obtiene el modelo buscado que permitirá clasificar cualquier nueva conexión. Durante la fase de monitoreo, sólo se envía el tren de paquetes de medida y se puede decidir si admitir o no la conexión antes de que ésta ingrese en la red.

Esta página ha sido intencionalmente dejada en blanco.

# Capítulo 5

## Estimador del estado de la red

### 5.1. Introducción

En este capítulo se describirá la variable  $X$  del modelo detallado anteriormente. Como se explicó en la introducción, el estado de una red inalámbrica no es fácil de estimar considerando mediciones locales en cada nodo que conforma la red, por lo tanto, se toma la estrategia de considerar un modelo estadístico a partir de mediciones de extremo a extremo.

Partiendo de la hipótesis de que los tiempos entre arribos de los paquetes de medida dan información del estado de la red, como se verá a continuación,  $X$  se estimará a partir de la distribución de probabilidad de la componente variable del retardo percibido por el tren de paquetes de medida.

### 5.2. Definición de $X$

Por cada paquete de medida se puede identificar el tiempo de arribo a la cola del enlace inalámbrico correspondiente ( $t^i$ ) y el tiempo de llegada al destino ( $t^o$ ). En la práctica esto se consigue mediante un timestamp en los paquetes, conjuntamente con cierta estabilidad de los relojes de los nodos de la red (estabilidad en la escala de los milisegundos).

Considerando:

- $D$ : delay del enlace
- $C_n(p, I)$ : capacidad libre del enlace correspondiente (en caso de adaptación de rate la capacidad varía con la probabilidad de colisión ( $p$ ) y con la interferencia ( $I$ ) externa)
- $P$ : tamaño del paquete de medida
- $V_n(p, I)$ : delay ocasionado por las retransmisiones

## Capítulo 5. Estimador del estado de la red

- $tq_n$ : tiempo de espera en la cola

Para el paquete  $n$ -ésimo se tiene que:

$$t_n^o - t_n^i = \frac{P}{C_n(p, I)} + D + V_n(p, I) + tq_n \quad (5.1)$$

Algunos factores en (5.1) son constantes, por lo tanto se puede expresar la ecuación anterior dividiéndola en dos términos: uno constante y el otro variable según  $p$  e  $I$ .

$$t_n^o - t_n^i = K + K_n(p, I) \quad (5.2)$$

Tomando en cuenta dos paquetes consecutivos ( $n-1$  y  $n$ ) donde se cumple  $t_n^o - t_n^i = K + K_n(p, I)$  y  $t_{n-1}^o - t_{n-1}^i = K + K_{n-1}(p, I)$ , se obtiene:

$$K_n = K_{n-1} + (t_n^o - t_{n-1}^o) - (t_n^i - t_{n-1}^i) \quad (5.3)$$

Aplicando la ecuación (5.3) recursivamente se obtiene:

$$K_n = K_0 + \sum_{j=1}^n [(t_j^o - t_{j-1}^o) - (t_j^i - t_{j-1}^i)] \quad (5.4)$$

La relación 5.4 permite estimar la distribución de probabilidad de la componente variable del retardo sólo usando los tiempo de arribos y partidas de los paquetes de medida. Es importante señalar que  $K_0$  puede setearse a cero considerando la secuencia a partir de dos paquetes cuyo tiempo entre arribos sea igual (o lo más cercano) al tiempo entre partidas. Este sería el caso de un estado de la red con escasa carga, bajas colisiones y baja interferencia.

El análisis anterior se puede generalizar a una red de varios saltos, donde el retardo también tendrá una componente fija y una variable. Esta última es dada por la variabilidad de los diferentes saltos (en cada enlace se tiene una probabilidad de colisión, una cola, etc). Por lo anterior, se puede decir que la ecuación 5.4 es independiente de la cantidad de saltos del camino inalámbrico.

Cabe destacar que  $K_n$  sólo depende de las diferencias de los tiempos en puntos con el mismo reloj. Otro punto a mencionar es que en ningún lugar se usa el hecho de que los tiempos entre partidas de los paquetes de medida sean constantes, por lo tanto, el análisis realizado es válido para cuando éstos siguen una distribución cualquiera.

En las figuras siguientes se muestra el comportamiento de la distribución empírica de  $K_n$  para distintos estados de la red. En particular, las figuras 5.2 y 5.3 fueron realizadas en una WLAN 802.11 (sin diferenciación de clases de tráfico), en cambio 5.4 muestra un ejemplo en una red donde las estaciones implementan 802.11e.

## 5.2. Definición de $X$

Para las figuras 5.2 y 5.3 se usó la topología mostrada en la figura 5.1 para el caso de tráfico en condiciones de saturación (todos los nodos clientes tienen tráfico para enviar en todo momento). En la 5.2 se observa el efecto de cambiar la cantidad de nodos ( $m$ ) presentes en la red inalámbrica (en las cuatro simulaciones mostradas la distancia entre el RN y el NodoX<sup>1</sup> se mantiene constante). Se puede observar que cuando  $m$  crece, el valor de la media de  $K_n$  crece también. Por otro lado, en la 5.3, la distancia entre el NodoX el RN ( $d$ ) es la que varía (siendo la cantidad de nodos clientes ( $m$ ) constante en todas las simulaciones). También se observa el efecto que tiene el cambio en la distancia en la media y varianza de  $K_n$ .

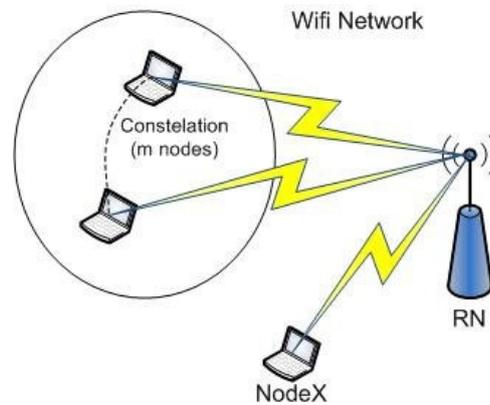


Figura 5.1: WLAN, escenario usado para mostrar el comportamiento de  $K_n$  al variar distintas condiciones del estado de la red.

En el caso con QoS, las pruebas fueron realizadas en una topología similar a la mostrada en la figura 5.1 pero en este caso los nodos no cambian sus posiciones respecto al nodo de referencia. Los distintos estados de la red se logran cambiando la cantidad de flujos de cada clase. Se consideraron dos clases de servicio: alta prioridad o clase 1 (ejemplo VoIP) y baja prioridad o clase 2 (ejemplo descargas de archivos). En el ejemplo particular de la figura 5.4 se muestra como se comporta el estimador  $K_n$  al variar la cantidad de flujos de clase 2 ( $b$ ) presentes en la red manteniendo constante la cantidad de flujos de clase 1 ( $a$ ).

El comportamiento de las curvas coincide con el esperado al analizar el modelo representado por la ecuación 5.4. Por ejemplo, al aumentar la cantidad de nodos presentes en la red aumentan las colisiones y por lo tanto aumentan las retransmisiones (reflejado en la ecuación 5.1). Esto provoca que aumente la parte variable del retardo lo que se ve traducido en un aumento en la media de  $K_n$ . Un comportamiento similar en el retardo variable ocurre al aumentar la distancia. El fenómeno se debe a que la SNR disminuye (aumenta la interferencia) al aumentar la distancia y por lo tanto baja la tasa de datos de capa física, ocasionando un

---

<sup>1</sup>NodoX es el nodo de la nueva conexión, encargado de enviar también el tren de paquetes de medida

## Capítulo 5. Estimador del estado de la red

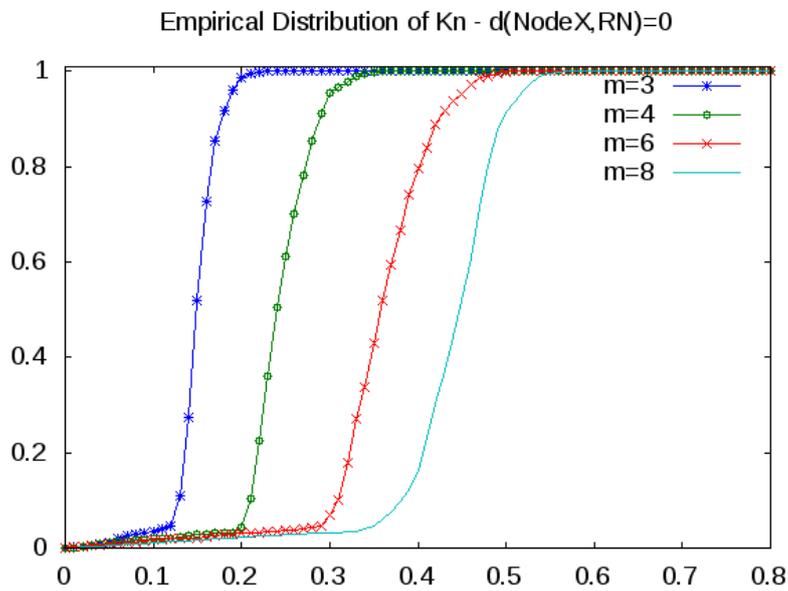


Figura 5.2: Distribución empírica de  $K_n$  para diferentes estados de la red (varía  $m$ )

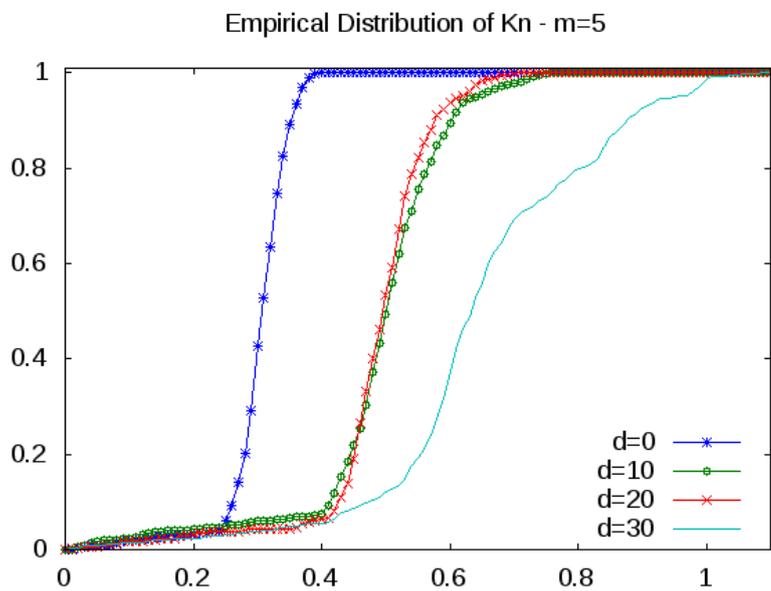


Figura 5.3: Distribución empírica de  $K_n$  para diferentes estados de la red (varía  $d$ ).

aumento en el retardo experimentado por los paquetes de medida. En este caso el fenómeno se traduce en un aumento de la media y un aumento de la varianza de  $K_n$ .

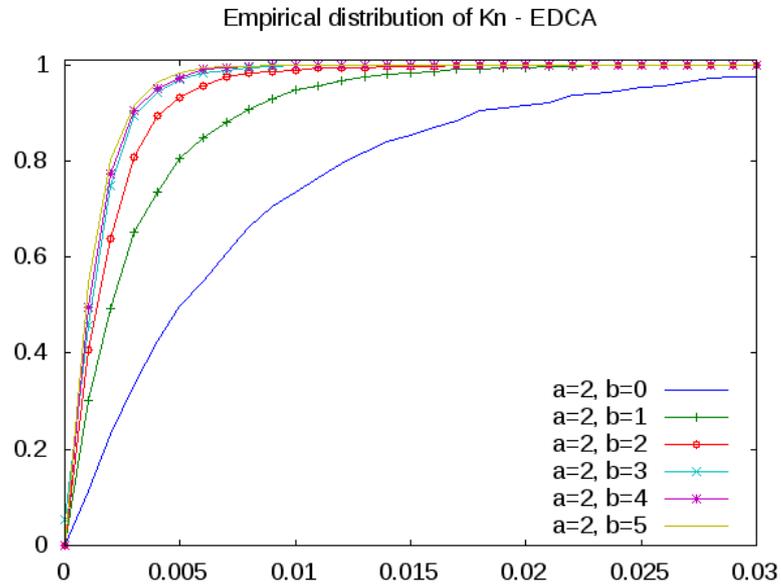


Figura 5.4: Distribución empírica de  $K_n$  para diferentes estados de la red (QoS). Los valores de  $a$  y  $b$  representan la cantidad de conexiones de clase 1 (alta prioridad) y 2 (baja prioridad) respectivamente.

Por otra parte, en las simulaciones correspondientes a la figura 5.4, como se explicará más adelante, los paquetes de medida se enviaron por la clase 1. Se observa en todas las curvas que, con probabilidad alta se tienen retardos variables muy pequeños (esto se debe en gran parte a la alta probabilidad de acceder al medio que tienen los paquetes de prueba y a la poca cantidad de flujos presentes en la red).

Luego de analizar los ejemplos mostrados en las figuras, resulta bastante lógica la idea de que el estimador del estado de la red ( $X$ ) se va a obtener a partir de algún estadístico de  $K_n$  como por ejemplo: media, varianza, combinación de ambas medidas, etc.

Esta página ha sido intencionalmente dejada en blanco.

## Parte III

# VALIDACIÓN DE LA METODOLOGÍA



# Capítulo 6

## Simulaciones

### 6.1. Introducción

En los capítulos anteriores se explicó en detalle la metodología, ahora llega el turno de la validación de la misma. El proceso de validación se realizó en una primera instancia mediante simulaciones ejecutadas en el simulador ns-2<sup>1</sup>. Una vez verificada mediante las simulaciones se realizaron pruebas experimentales en redes de laboratorio (ver Capítulo 7).

Primeramente se mencionarán generalidades del simulador utilizado, en especial algunas características de cómo es la implementación de 802.11. Luego, el capítulo se concentrará en la parte de simulaciones, se mostrarán los resultados obtenidos más representativos para los dos casos de aplicación de la metodología: estimación de parámetros de QoS (en particular del throughput) y decisiones de control de admisión.

### 6.2. Simulador ns-2

Se trata de un simulador de redes de eventos discretos extremadamente popular en el ámbito académico. Este simulador se ha ido desarrollando estos últimos años desde que 1989 empezara como una variante del simulador REAL. Sigue una filosofía de desarrollo abierta (licencia GPL), lo que ha contribuido a su crecimiento, convirtiéndose en una de las herramientas más completas para el estudio de redes. Se puede ejecutar en diferentes plataformas como Windows o Linux y además está disponible gratuitamente en la página oficial [57].

Consta de un núcleo principal escrito en lenguaje C++ y una interfaz de

---

<sup>1</sup>ns-2: simulador de redes más empleado por los investigadores que publican trabajos sobre temas relacionados con 802.11. Se probaron otros simuladores: ns-3.9 y OMNeT++, pero se eligió el ns-2 por ser el que mejor se adaptaba a los resultados teóricos esperados (en particular en las pruebas considerando QoS (802.11e)).

## Capítulo 6. Simulaciones

usuario OTcl<sup>2</sup>. Como capa de interfaz entre C++ y OTcl se usa TclCL. Permite trabajar tanto en redes cableadas como en redes inalámbricas utilizando un gran número de protocolos de capa de aplicación (CBR, FTP, Telnet, etc) y de transporte (TCP y UDP).

En el desarrollo de esta tesis se comenzó trabajando con la versión 2.34 (última en ese momento) analizando la metodología en redes 802.11b/g. A la hora de simular redes 802.11e no se tuvo otra opción que migrar a la versión 2.28 para poder utilizar uno de los parches disponibles.

### 6.2.1. Implementación de 802.11

La versión básica del ns implementaba la función DCF pero era considerada muy pobre porque obviaba puntos importantes del estándar.

La librería *dei802mr* [58] incluida a partir de la versión 2.33 del simulador, incorpora nuevos modelos físicos con diferentes rates de transmisión para 802.11b/g (modo b: 1/2/5,5/11 Mbps, modo g: 6/9/12/18/24/36/48/54 Mbps). Además lleva a cabo un cálculo más preciso del SINR (Signal to Interference-plus-Noise Ratio), del PER (Packet-Error-Rate) a partir de las curvas PER vs SINR vs el tamaño del paquete (proporcionadas para 802.11b/g y modificables por el usuario). Otra contribución de esta librería es que la potencia de ruido es configurable en los scripts Tcl.

El simulador permite la utilización de tres modelos de propagación: Free-space, Shadowing y Two-ray ground reception. Este último considera para el cálculo de potencia recibida el aporte de dos rayos: el directo y el reflejado en el suelo, obteniendo mejores resultados cuando se trata de enlaces largos. Para el caso de las simulaciones de redes de área local se tomó el modelo Free-space, en cambio, en las simulaciones realizadas considerando topologías de uso común en redes rurales (por ejemplo enlaces de larga distancia) se eligió Two-ray ground reception.

### 6.2.2. Extensión a 802.11e

Los módulos oficiales de ns-2 no soportan esta versión del estándar. Se han encontrado diferentes implementaciones externas ([59], [60], [61]), aunque ninguna para la última versión de ns-2. Para este trabajo se eligió utilizar el proyecto de la Universidad Técnica de Berlín [59] (parche para ns-2.28). Éste da soporte básico para EDCA, aunque no implementa adaptación de velocidad, asociación, autenticación ni gestión de energía. Se basa en la implementación existente de DCF, pero a su vez corrige bugs y carencias de dicha versión. Dado nuestro interés por las topologías a utilizar en redes rurales en donde la adaptación de velocidad no es recomendada, este desarrollo, muy usado en otros trabajos académicos (ver por

---

<sup>2</sup>OTcl (Object Tcl) es la extensión del lenguaje Tcl orientada a objetos.

ejemplo [62]), resulta suficiente para el análisis deseado.

Los detalles de la implementación y verificación del funcionamiento del parche elegido se encuentran en el Technical Report TKN-03-19 [63].

### Script de simulación

Son pocos los cambios necesarios para transformar un script 802.11 a uno 802.11e, este fue otro de los motivos que llevó a continuar trabajando con ns-2 y no continuar investigando otros simuladores. Además es posible reutilizar, casi en su totalidad, los scripts de procesamiento de trazas construidos para las simulaciones con la versión 2.34. A continuación se detallan los cambios necesarios para la migración de un script realizado en la versión 2.34, que no tiene distinción de clases en capa MAC, a uno en la versión 2.28 utilizando EDCA como mecanismo de acceso al medio.

En primer lugar hay que indicar el tipo de MAC y de encolado para que soporte colas con prioridad, esto se consigue modificando las variables *val(mac)* y *val(ifq)* del script OTcl con las siguientes dos líneas de código:

- set val(mac) Mac/802\_11e
- set val(ifq) Queue/DTail/PriQ

Una vez creados los agentes de tráfico, se debe indicar la prioridad que tendrá el tráfico generado por cada uno. Esto se indica con la variable *prio\_* que puede tomar los valores entre 0 y 3, siendo 0 el de mayor prioridad. En EDCA, cada nodo tendrá cuatro colas droptail, una para cada clase de acceso. A continuación se muestra, a modo de ejemplo, como configurar prioridad 0 a un agente de tráfico UDP:

- set udp\_(1) [new Agent/UDP]
- \$udp\_(1) set prio\_ 0

Es posible también, modificar las prioridades de las colas cambiando los valores del archivo *priority.tcl* (ubicado en ns-x/mac/802\_11e/). Para que las modificaciones tengan efecto se debe recompilar el simulador. Las prioridades se logran cambiando los extremos de la ventana de contienda, los tiempos de espera del protocolo de acceso al medio y el valor del TXOP.

En el caso del trabajo realizado enmarcado en la tesis se utilizaron los valores por defecto que trae el simulador para los distintos parámetros de las clases de acceso. A modo de ejemplo, en el Apéndice B se muestra un script completo para simular una red 802.11e.

## 6.3. Estimación de Throughput en una red 802.11

### 6.3.1. Introducción

Es el turno de mostrar los resultados obtenidos para la validación de la metodología cuando es usada para estimar parámetros de QoS.

Para la prueba se consideró una topología del tipo WLAN conformada por un nodo principal (RN) y cierta cantidad de nodos fijos aleatoriamente distribuidos en la región de cobertura del nodo RN (ver Figura 6.1). Para este caso se configuró el protocolo IEEE 802.11g y ARF (Auto Rate Fallback [39] [64]) como algoritmo de adaptación de rate. Todas las simulaciones mostradas en esta sección fueron realizadas usando la versión 2.34 del ns-2 ya que no se consideraron distintas clases de tráfico.

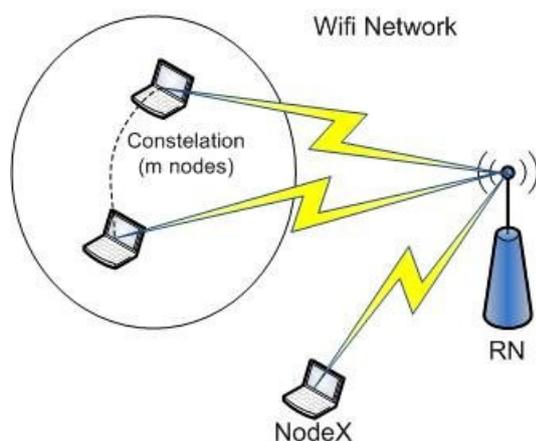


Figura 6.1: Topología utilizada para testear la metodología para el caso de predicción de throughput (en condiciones de tráfico saturadas y no saturadas).

A continuación se presentarán los resultados obtenidos para dos escenarios: modelo de tráfico saturado y no saturado.

Tal como se vio en el capítulo 4, para lograr entrenar el sistema y luego verificar el modelo es necesario contar con un gran número de muestras para así realizar un buen aprendizaje estadístico. Cada muestra se obtiene a partir de una corrida de simulación. Con el fin de entrenar el sistema emulando varios estados de la red, en cada corrida se cambió la distancia entre el NodeX y el RN así como también la cantidad de nodos presentes en la red y sus ubicaciones.

### 6.3.2. Condiciones Saturadas de Tráfico

Se desea estimar el máximo throughput que obtendrá una conexión inalámbrica entre dos puntos (NodoX y RN) dada ciertas condiciones de stress en la red. En este caso se utiliza un modelo donde todos los nodos tienen paquetes para enviar en todo momento (saturación).

Características generales de cada simulación:

- Topología: RN +  $m$  nodos clientes + NodoX (ver figura 6.1).
- Tipo de tráfico: Todos los nodos (inclusive el NodoX) generan tráfico del tipo CBR/UDP uplink<sup>3</sup> (Rate=20 Mbps<sup>4</sup> y Tamaño de paquete=1500 bytes).
- Tren de paquetes de medida: UDP, tamaño: 10 bytes (payload), tiempo entre paquetes= 5 ms (\*)
- Las posiciones de los nodos se obtienen utilizando un proceso de Poisson dentro de un área circular con centro en el RN<sup>5</sup>.

(\*) Es importante resaltar que existe un compromiso entre el tamaño de los paquetes de medida y el tiempo entre partidas dado que el objetivo es no afectar el desempeño de la red. Se hicieron varias pruebas y se llegó a que utilizando los valores mencionados (10 bytes y 5 ms respectivamente) es suficiente para lograr muy buenos resultados sin afectar significativamente el sistema.

En cada corrida de simulación se pueden diferenciar las tres etapas explicadas en el **Procedimiento A** (donde  $t$  representa el tiempo de simulación).

**Procedimiento (A):**

- Etapa 1 ( $t = 0$  ,  $t = 50$ ): Primeramente se sortean las posiciones de los nodos clientes. Los  $m$  nodos de la red comienzan a enviar paquetes de acuerdo a las características mencionadas. Continúan cursando tráfico hasta que el sistema llega a un estado estacionario. Estos nodos siguen transmitiendo hasta el final de la simulación ( $t = 140$ ).
- Etapa 2 ( $t = 50$  ,  $t = 100$ ): El NodoX envía tráfico CBR/UDP y es en este momento donde se mide el throughput alcanzado por dicha conexión. Como se está en condiciones de saturación, el valor medido significa el máximo throughput que podría lograr una nueva conexión de ese tipo entre el NodoX y RN dado el estado de la red definido en la Etapa 1. En  $t = 100$  el NodoX deja de transmitir. Entre  $t = 100$  y  $t = 130$  se espera a que la red llegue nuevamente al estado estacionario.

<sup>3</sup>uplink: desde los nodos clientes con destino el RN. Es cuando tiene sentido el fenómeno de las colisiones y resulta más interesante el estudio.

<sup>4</sup>El rate grande garantiza el estado de saturación en todo momento independiente de la cantidad de nodos presentes en la red

<sup>5</sup>Se usaron radios de: 10 metros, 30 metros y 60 metros.

## Capítulo 6. Simulaciones

- Etapa 3 ( $t = 130$ ,  $t = 140$ ): El NodoX envía el tren de paquetes de medida con las características mencionadas. Se registran los tiempos de arribos y partida de dichos paquetes. Con estos tiempos se construye  $K_n$  de acuerdo a la ecuación 5.4.

Por cada simulación se obtiene como salida:

- el **throughput** de aplicación alcanzado por la conexión entre el NodoX y RN
- el **estimador del estado de la red** ( $X$ ) para la ruta de la conexión anterior

*Observación:* El cálculo del throughput se realizó considerando la cantidad de bytes recibidos por el nodo destino y el tiempo total utilizado en la transmisión de dicha carga (la medida se realizó tomando los paquetes recibidos entre  $t = 70$  y  $t = 100$  para evitar el transitorio que se genera al agregar un nuevo flujo en la red). Los intervalos de tiempo considerados son bastante conservadores.

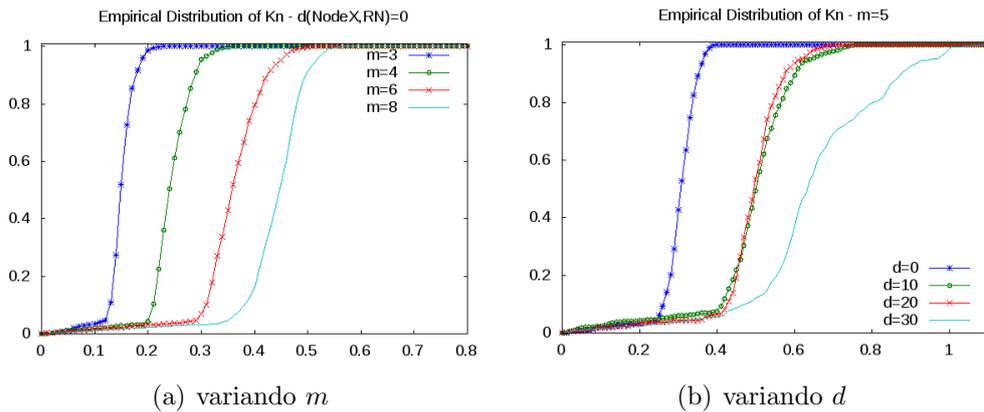


Figura 6.2: Distribuciones empíricas de  $K_n$  para distintos escenarios. Variando la cantidad de clientes  $m$  o variando la distancia  $d$  entre el NodoX y RN.

Se realizaron cuatro conjuntos de simulaciones: uno para entrenar el sistema y los demás para verificar el modelo conseguido. Cada conjunto se obtuvo cambiando la topología (considerando distintas áreas para realizar el sorteo de las posiciones de los nodos clientes). En cada conjunto se realizaron  $m * d$  simulaciones, siendo  $m$  el número de nodos en la red y  $d$  la distancia entre NodoX y RN (rangos usados:  $m = 1, \dots, 9$  y  $d = 0, \dots, 40$ ).

Dadas las figuras 6.2 y una vez realizado el análisis correspondiente se observó que sólo utilizando la media de  $K_n$  como  $X$  se obtienen buenas predicciones. Se verificaron otros valores para  $X$  como la varianza de  $K_n$ , combinación entre media y varianza, etc, pero ninguno mejoraba sustancialmente la precisión alcanzada con la media.

### 6.3. Estimación de Throughput en una red 802.11

En la tabla 6.1 se muestran los resultados obtenidos al aplicar SVM. En las figuras 6.3, 6.4 y 6.5 se muestra el throughput simulado y su estimación al aplicar el modelo obtenido aplicando SVM para los tres conjuntos de verificación. Cada conjunto se diferencia en el área (círculo de radio  $R$  con centro en el RN) considerada para sortear las posiciones de los nodos.

Tabla 6.1: Resultados de SVM utilizando la media de  $K_n$ .

Mean squared error	0.24232
Squared correlation coefficient	0.878246

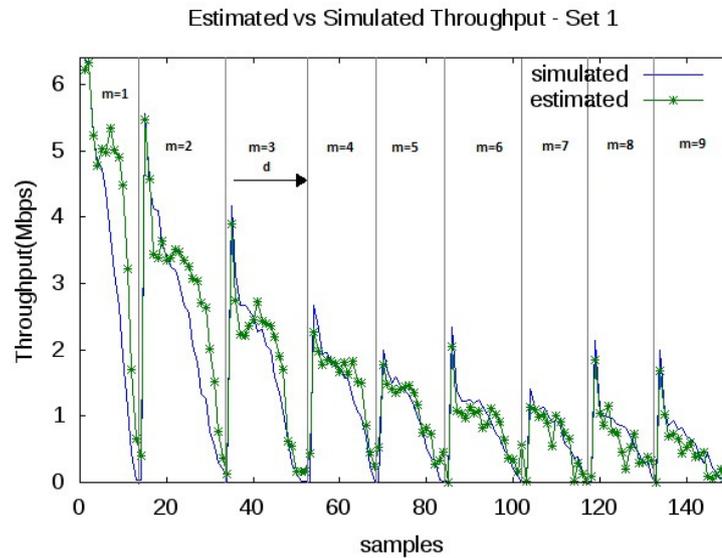


Figura 6.3: Conjunto 1 de verificación: Predicciones usando sólo la media de  $K_n$  como estimador del estado de la red (condiciones de tráfico saturadas).  $R = 30$  metros.

En cada conjunto, para cada valor de  $m$ , se sortea una vez la posición de los nodos clientes, luego se realizan tantas corridas de la simulación como valores posibles se consideren para la distancia entre los nodos de interés. Al aumentar  $d$  (siendo la cantidad de nodos y sus posiciones constante) el throughput disminuye como es de esperar. Al aumentar  $m$ , para un mismo valor de  $d$ , es también esperable que el throughput percibido por la nueva conexión disminuya como efectivamente ocurre. Esta manera de realizar las simulaciones explica el por qué de la forma de las curvas mostradas en las figuras 6.3, 6.4 y 6.5.

Es importante resaltar que entrenando con sólo 100 muestras (que corresponden a 100 duplas  $(X_i, Y_i)$ , siendo  $X_i$  el estimador para un cierto estado de la red y  $Y_i$  el throughput que se lleva la nueva conexión entre el RN y NodoX) se pueden obtener muy buenos resultados.

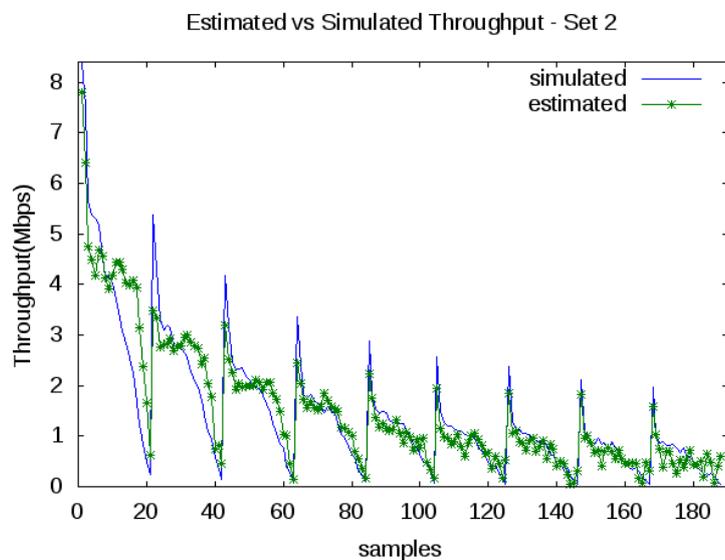


Figura 6.4: Conjunto 2 de verificación: Predicciones usando sólo la media de  $K_n$  como estimador del estado de la red (condiciones de tráfico saturadas).  $R = 10$  metros.

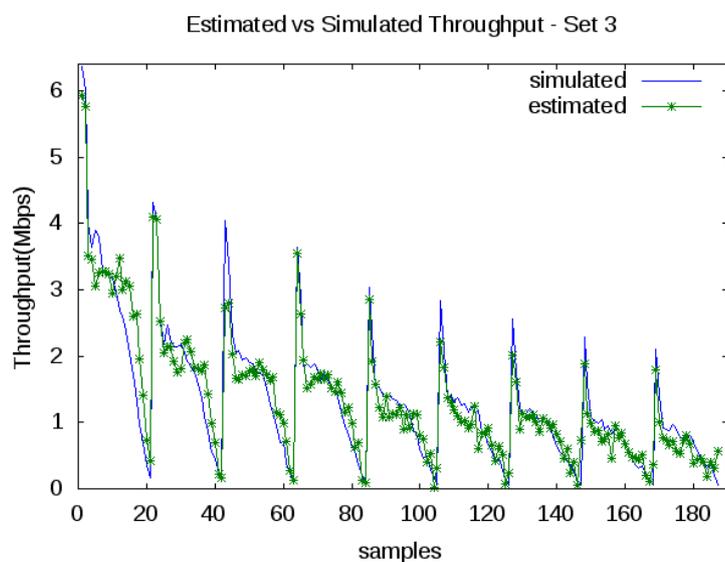


Figura 6.5: Conjunto 3 de verificación: Predicciones usando sólo la media de  $K_n$  como estimador del estado de la red (condiciones de tráfico saturadas).  $R = 60$  metros.

### 6.3.3. Condiciones NO Saturadas de Tráfico

En esta sección se tiene como objetivo la estimación del throughput máximo obtenido por la conexión entre el NodoX y el RN cuando la red no se encuentra exigida a su máxima capacidad. Por tal motivo, se consideró que el tráfico generado por los nodos simula pequeñas descargas o navegación por Internet. Este tipo de

### 6.3. Estimación de Throughput en una red 802.11

tráfico se modeló utilizando procesos exponenciales ON/OFF. Por simplificación se apagó el mecanismo de adaptación de rate (ARF).

Características generales de cada simulación:

- Topología: RN +  $m$  nodos clientes + NodoX (misma arquitectura que para el caso saturado)
- Tipo de tráfico: Todos los nodos (salvo el NodoX) generan tráfico del tipo ON-OFF Exponencial usando UDP como protocolo de transporte. El tráfico es uplink con las siguientes características: burst time= 0,5 s, idle time= 0,5 s, mean rate= 500 kbps, Packet Size= 1500 bytes.
- Tren de paquetes de medida: UDP, tamaño: 10 bytes (payload), tiempo entre paquetes= 5 ms
- Las posiciones de los nodos se obtienen utilizando un proceso de Poisson considerando un área circular con centro en el RN.
- El NodoX genera tráfico CBR/UDP uplink (Rate=20 Mbps Packet Size= 1500 bytes). Éste se encarga de saturar la red, así se obtiene el máximo throughput dadas las condiciones de la red determinadas por la cantidad de nodos clientes y sus posiciones.
- Physical Layer Rate= 12 Mbps (fijo).

Se procede de la misma forma que en el caso anterior: incrementando  $m$  (la cantidad de nodos presentes en la red) y variando la distancia entre NodoX y RN ( $d$ ). Cada simulación puede descomponerse en los mismos pasos mencionados en el **Procedimiento (A)** detallado anteriormente.

En la Figura 6.6 se muestra la distribución empírica de  $K_n$  para distintos estados de la red (variando cantidad de clientes o distancia de los nodos al RN, etc). Como se puede observar, el comportamiento es bastante distinto al del caso saturado (figuras 6.2). En el saturado, con probabilidad baja se tienen valores de retardo variable pequeños, en cambio, en este caso se tiene el efecto contrario (es coherente dada la poca carga en la red comparada cuando la red está al máximo de su capacidad como es el caso saturado).

Se verificó que utilizando sólo la media de  $K_n$  como variable  $X$  se obtiene una estimación pobre del parámetro de QoS deseado. Dicha estimación mejora agregando la varianza, aunque todavía existen puntos donde la estimación no es lo suficientemente buena. Luego de varias pruebas se concluyó que utilizando los siguientes tres parámetros como estimadores del estado del enlace ( $X$ ) se logra una buena precisión en la estimación buscada:

- Media y Varianza del estimador  $K_n$

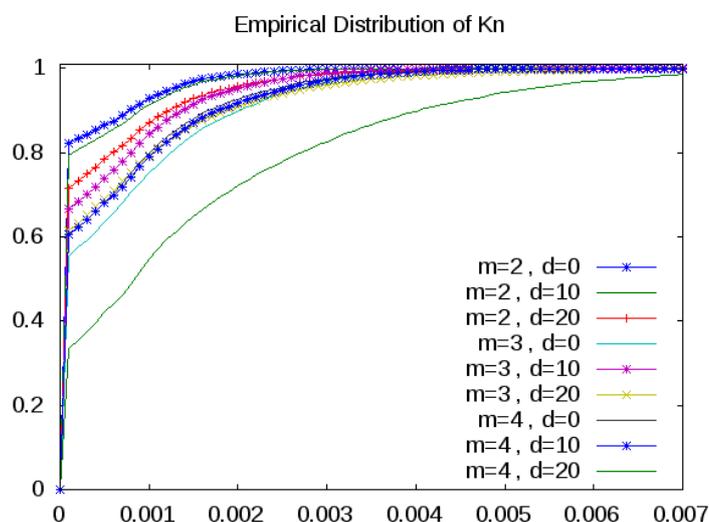


Figura 6.6: Distribución empírica de  $K_n$  para distintos estados del enlace inalámbrico.

- Valor de la distribución empírica de  $K_n$  en 0 (mínimo delay variable percibido en las condiciones de red correspondientes). Como se observa en la figura 6.6 el valor de  $K_0$  discrimina bastante bien. Esto no ocurriría en el caso saturado.

En este caso se realizaron dos conjuntos de simulaciones: uno para entrenar el sistema y obtener el modelo usando SVM, y el otro para su verificación. A continuación se muestran los resultados obtenidos al aplicar SVM junto con la gráfica mostrando el throughput simulado y el estimado mediante el modelo (ver Tabla 6.2 y Figura 6.7).

En la figura 6.7 se muestra el throughput para cada experimento del conjunto de validación. Se usaron 81 valores de throughput para verificar el modelo, los primeros nueve son simulaciones realizadas con  $m = 1$  variando la distancia entre el RN y el NodoX. Los segundos nueve valores son con  $m = 2$ , y así sucesivamente hasta  $m = 9$ .

Tabla 6.2: Resultados SVM.

Mean squared error	0.321387
Squared correlation coefficient	0.866621

### 6.3. Estimación de Throughput en una red 802.11

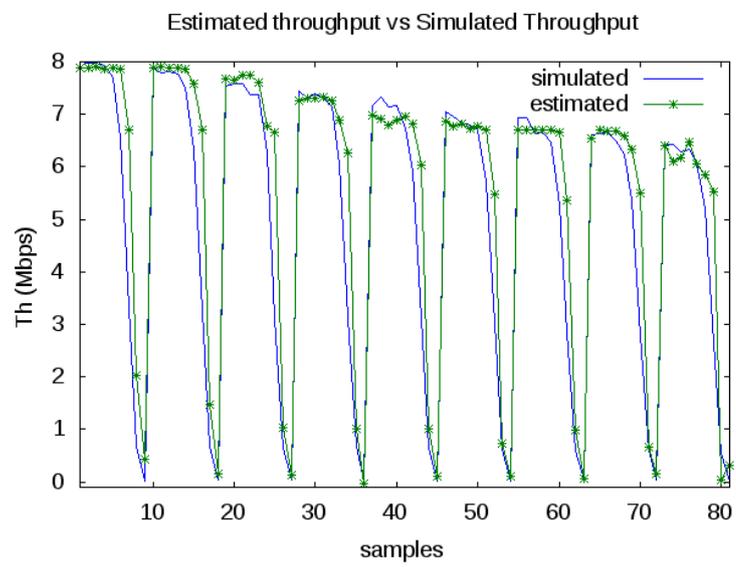


Figura 6.7: Throughput Simulado y Estimado para el caso de condiciones no saturadas de tráfico usando media, varianza y delay variable mínimo como  $X$ .

## 6.4. Herramientas para decisiones de Control de Admisión para una red 802.11e

### 6.4.1. Introducción

En esta sección se detallarán las pruebas de validación en el caso de utilizar la metodología para tomar decisiones de control de admisión en redes 802.11e. Se buscará decidir si aceptar o no un nuevo flujo, procurando garantizarle ciertos niveles de QoS, dependiendo del estado de la red.

Es importante mencionar que todo lo realizado a continuación para 802.11e es compatible con 802.11n. Esta nueva extensión del estándar, a pesar de permitir MIMO (multiple-input/multiple-output) y velocidades de capa física mucho mayores que las versiones b/g (las más usadas actualmente), obliga a los fabricantes a implementar el mecanismo de acceso al medio definido en 802.11e.

En este caso, se eligieron para el testeo dos topologías: *(I)* escenario compuesto por un nodo de referencia (RN o AP) y cierto número de nodos clientes, *(II)* un enlace punto a punto entre dos nodos WiFi. La *(I)* puede verse como una red de acceso residencial con EDCA como mecanismo de acceso al medio (802.11e). La *(II)* es representativa de una de las topologías más usadas en entornos rurales aislados.

Todas las simulaciones de esta sección fueron realizadas con ns-2.28, esto es, porque el parche existente que modela el funcionamiento de EDCA [59] está disponible para dicha versión del simulador <sup>6</sup>.

Tal como se explicó en el capítulo 4, por simplificación se consideraron sólo dos clases de servicio (una de mayor prioridad que otra). En la tabla 6.3 se listan las dos clases junto con los parámetros de acceso al medio utilizados.

Tabla 6.3: Parámetros usados para las dos clases de servicio (EDCA).

Clase	AC	CWmin	CWmax	AIFSN	TXOPLimit(s)
Clase 1	AC_VO	7	15	2	0.003264
Clase 2	AC_BE	31	1023	7	0

Las primeras preguntas que nos planteamos fueron: En este caso que tenemos

---

<sup>6</sup>La instalación del ns-2.28 junto con la configuración del parche mencionado llevó más tiempo del previsto dado que se trata de una versión bastante antigua del simulador. Se tuvo que trabajar sobre un sistema operativo con un kernel “viejo” (Ubuntu 9.04 kernel 2.6.28-19). Además de realizar varias modificaciones en algunos archivos del ns para lograr la instalación [65], se tuvo que trabajar con gcc-3.3/g++-3.3. Un punto a destacar es que los scripts realizados para ns-2.34 se reutilizaron casi en su totalidad para las simulaciones en ns-2.28 (salvo por los agregados correspondientes al manejo de QoS).

#### 6.4. Herramientas para decisiones de Control de Admisión para una red 802.11e

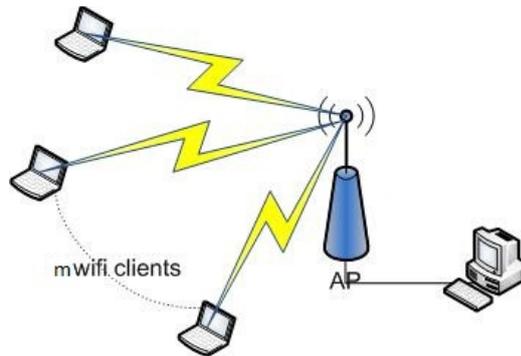


Figura 6.8: Topología I: WLAN 802.11e

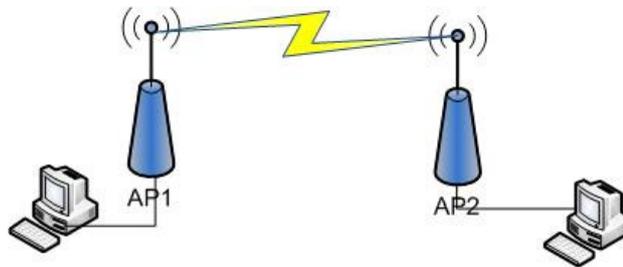


Figura 6.9: Topología II: enlace punto a punto

más de una clase de servicio, ¿Por qué clase mandar los paquetes de medida? ¿Es necesario mandar dichos paquetes por todas las clases de servicio para tener una estimación total del canal?

Luego de realizar varias pruebas se verificó que no es necesario enviar el tren de paquetes de medida por todas las AC consideradas sino que enviándolo sólo por una de ellas ya es suficiente para poder estimar el estado de la red y por lo tanto realizar decisiones de control de admisión para nuevas conexiones. Independientemente de por cuál clase se realice el envío de los paquetes de medida, lo que se obtiene es la estimación del estado de la red desde un punto de referencia.

Dado un estado de la red  $e_i$ , si el tren de paquetes es enviado por la clase 1 aplicando la ecuación 5.4 obtengo  $K_{n_i}^1$ , de lo contrario obtengo  $K_{n_i}^2$ . A partir de los  $K_{n_i}^j$  puedo elegir los estimadores del estado de la red propiamente dichos  $X_i^1$  y  $X_i^2$ . Al entrenar el sistema, lo que hago es obtener la relación entre el estado de la red y el parámetro de QoS deseado ( $Y$ ). El estudio se basa en que para cierto valor de  $Y_i$  existen  $\phi_1$  y  $\phi_2$  que cumplen:  $Y_i = \phi_1(X_i^1) + \epsilon$  y  $Y_i = \phi_2(X_i^2) + \epsilon$ . O sea, que sin importar de que punto de vista estoy mirando el estado de la red, a los efectos buscados, resulta indistinto mandar los paquetes de medida por la clase 1 o por la clase 2. Lo anterior es generalizable para  $N$  clases de tráfico.

Primero, lo que parecía más razonable era utilizar la clase 2 para el envío del tren de paquetes de medida dado que al tratarse de una clase con menor prioridad,

## Capítulo 6. Simulaciones

se lograría una menor inferencia en el desempeño de la red al momento de realizar el monitoreo. Una vez corridas las pruebas para validar la hipótesis anterior, se constató que en casos en que la red se encuentra cercana a su saturación, los flujos de clase 2 presentan una alta probabilidad de pérdidas lo que hace que se tengan pocas muestras (tiempos de arribo) para lograr estimar el estado de la red si el monitoreo se hace en un intervalo de tiempo razonable. Por lo anterior se decidió probar con la clase 1 para el envío de dichos paquetes y los resultados fueron muy buenos.

### 6.4.2. Topología I

A continuación se presentan dos escenarios: *A* y *B*. En *A* se considera el caso cuando la conexión a clasificar es de clase 1 y en *B* es de clase 2. En cada corrida de simulación se cambia la cantidad de flujos presentes en la red para emular distintos estados de carga (distintos niveles de utilización del aire). Se modeló el tráfico de todos los nodos como CBR/UDP y se estudió el sistema en un estado cercano a la saturación que es cuando el control de admisión tiene mayor sentido.

Características generales de cada simulación:

- Topología: AP +  $m$  nodos clientes + NodoX (la distancia promedio entre los nodos clientes y el AP es de 30 metros en promedio)
- Tipo de tráfico: Todos los nodos (incluido el NodoX) generan tráfico del tipo CBR/UDP uplink
- Tren de paquetes de medida: UDP, tamaño: 10 bytes (payload), tiempo entre paquetes = 10 ms (tráfico perteneciente a clase 1)
- En el escenario A, el NodoX genera tráfico de aplicación perteneciente a la clase 1 (caso contrario ocurre en el escenario B).

Similar a lo explicado en la sección anterior, cada simulación se descompone en las siguientes instancias (siendo  $t$  el tiempo de simulación):

- Parte 1 ( $t = 0$  ,  $t = 100$ ): Se sortea la cantidad de flujos de cada clase para ser enviados por las estaciones clientes ( $m$ ). Éstos comienzan a traficar en  $t = 0$  y continúan hasta el final de la simulación.
- Parte 2 ( $t = 100$  ,  $t = 150$ ): El NodoX realiza el envío del tráfico a clasificar (finaliza el envío en  $t = 150$ ). Si el tráfico es de clase 1 (escenario A) se registra el delay percibido, de lo contrario (escenario B) se determina el throughput alcanzado. Durante los siguientes segundos (hasta  $t = 180$ ) se espera a que la red vuelva a su estado estacionario.
- Parte 3 ( $t = 180$  ,  $t = 200$ ): Se envía el tren de paquetes de medida desde el NodoX al AP.

## 6.4. Herramientas para decisiones de Control de Admisión para una red 802.11e

### Resultados obtenidos para el Escenario A

Por cada simulación se obtiene como producto:

- el **delay** obtenido por la nueva conexión entre el NodoX y el AP (como es un flujo de clase 1 la idea es realizar su clasificación de acuerdo al delay)
- $X$ , el estimador del estado del enlace inalámbrico.

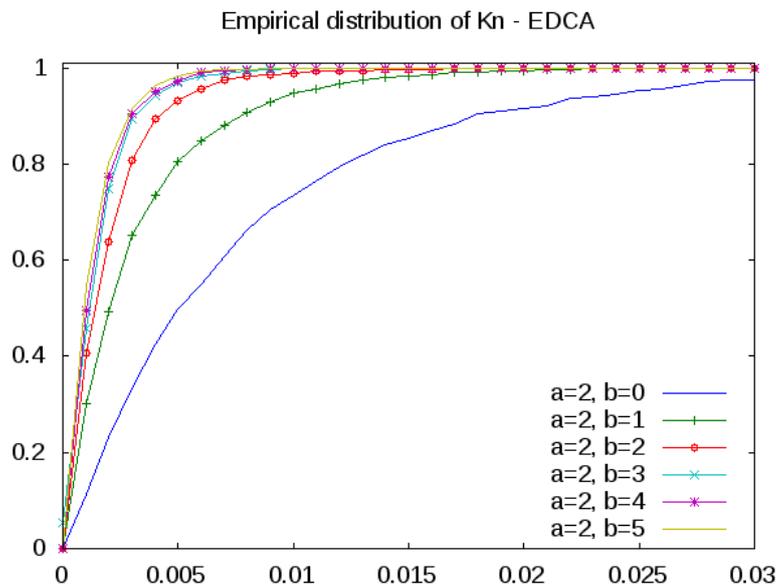


Figura 6.10: Distribución empírica de  $K_n$  para diferentes estados de la red. Los valores de  $a$  y  $b$  representan la cantidad de conexiones de clase 1 y 2 respectivamente (WLAN 802.11e).

Se construyó la variable  $Z$  definida en 4.3 eligiendo un umbral determinístico constante:  $maxDelay$ .  $Z = 1$  si el delay percibido es menor que  $maxDelay$ , de lo contrario vale 0. Con los tiempos de arribo y partida de los paquetes de medida y usando la ecuación 5.4 se determinó  $K_n$ . En este caso alcanza con considerar la media de  $K_n$  como  $X$  para obtener satisfactorios niveles de precisión en la clasificación.

La figura 6.10 muestra algunas distribuciones empíricas para este caso (se ve como varía la media de  $K_n$  al cambiar la cantidad de flujos clase 2 presentes en la red). En la figura 6.11 se muestra la relación existente entre el delay obtenido y el estimador del estado de la red elegido.

Se realizaron dos conjuntos de simulaciones, uno compuesto de 74 muestras para entrenar y el otro (73 muestras) para verificar. Usando SVM-C con  $(Z, X)$  se obtuvo una precisión del 94,5 %.

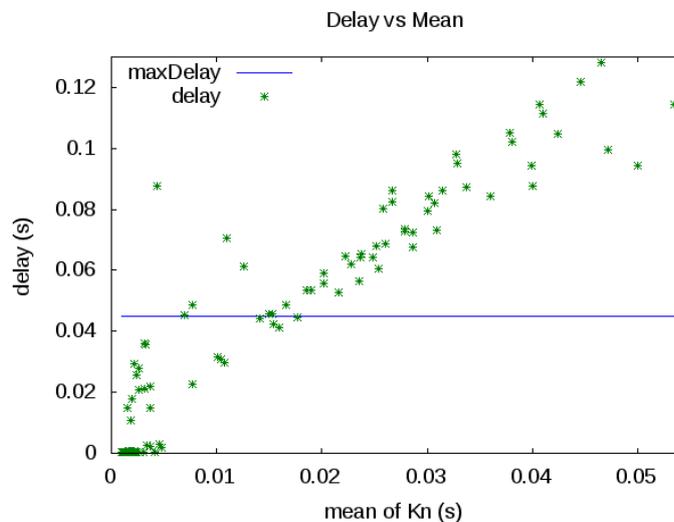


Figura 6.11: Delay obtenido para la nueva conexión de clase 1 vs Media de  $K_n$ . WLAN 802.11e, escenario A.

*Observación:* Cuando hablamos de precisión nos referimos a la proporción de las muestras de verificación que fueron clasificadas correctamente. En este caso, se clasificaron correctamente 69 muestras del total, lo que equivale al 94,5 %.

### Resultados obtenidos para el Escenario B

En este caso interesa clasificar la nueva conexión según el throughput alcanzado. La idea es verificar si la red permite por lo menos un mínimo de throughput ( $minTh$ ) cumpliendo con los requisitos de QoS de cierta aplicación.

En este caso, por cada simulación se obtiene:

- el **throughput** obtenido por la nueva conexión entre NodoX y RN (flujo de clase 2)
- $X$ , el estimador del estado del enlace inalámbrico.

Similar al caso anterior se construyó la variable  $Z$  comparando el throughput obtenido con el umbral elegido:  $minTh$ . Usando también la media de  $K_n$  como valor de  $X$  se obtuvo una precisión del 88,83 %. En la figura 6.12 se muestra la relación existente entre el throughput obtenido y el estimador del estado de la red elegido. También, en dicha figura se muestra el umbral de decisión usado.

## 6.4. Herramientas para decisiones de Control de Admisión para una red 802.11e

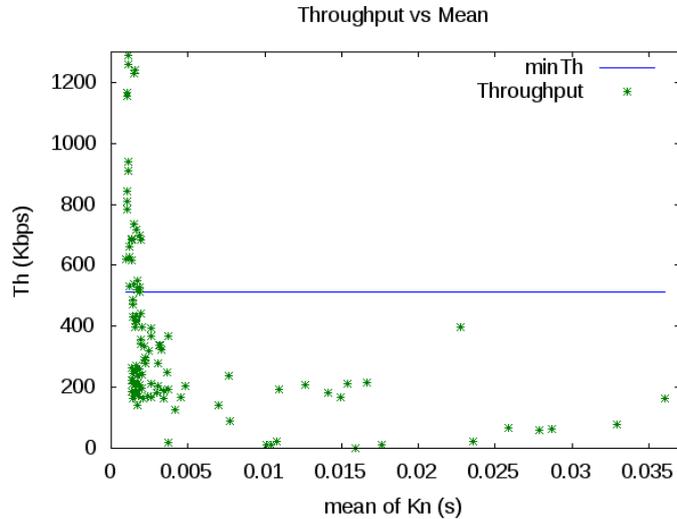


Figura 6.12: Throughput obtenido para la nueva conexión de clase 2 vs Media de  $K_n$ . WLAN 802.11e, escenario B.

### 6.4.3. Topología II

En este caso se utilizaron modelos de tráfico más realistas, para los flujos de clase 1 se utilizó un proceso exponencial ON/OFF utilizando UDP como protocolo de transporte. Por otro lado, los flujos de clase 2 se modelaron como tráfico FTP utilizando TCP en capa 4.

Características generales de cada simulación:

- Topología: enlace punto a punto entre dos nodos inalámbricos  $AP_1$  y  $AP_2$ , donde la distancia entre ellos es de 30 metros. Ambos nodos están conectados a dos PCs (ver Figura 6.9).
- Tipo de Tráfico: Ambas PCs generan tráfico de las dos clases de acceso consideradas. Análogamente a lo que se hizo para la topología anterior se estudiaron dos escenarios: en el escenario  $C$  la nueva conexión es de clase 1, y en el  $D$  el tráfico pertenece a clase 2.
- Tren de paquetes de medida: UDP, Packet Size= 10 bytes, Interval= 10 ms (clase 1)

Al igual que en lo mencionado para la Topología I, las simulaciones se dividen en las mismas tres etapas. Aplicando el mismo análisis que en casos anteriores se obtienen los siguientes resultados que demuestran buena precisión. En este caso se usaron 98 muestras para entrenar, y la misma cantidad para verificar.

En las figuras 6.14 y 6.15 se muestra el delay el throughput de la nueva conexión para los escenarios  $C$  y  $D$  respectivamente. En particular, para el escenario  $C$  se

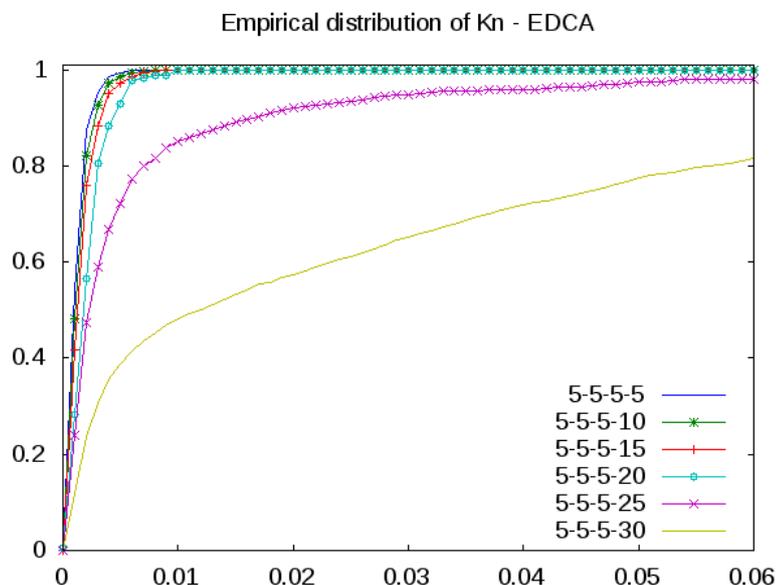


Figura 6.13: Distribución empírica de  $K_n$  para diferentes estados de la red. Las curvas se diferencian en la cantidad de flujos de clase 2 que se cursan desde el  $AP_1$  al  $AP_2$ . Todas las simulaciones se realizan con 5 flujos clase 1 desde el  $AP_1$  hacia el  $AP_2$ , 5 de clase 1 en sentido contrario y 5 de clase 2 desde el  $AP_2$  al  $AP_1$ .

Tabla 6.4: Resumen de resultados al aplicar SVM considerando como estimador del estado de la red la media de  $K_n$ .

Escenario	Descripción	Precisión (SVM-C)
Escenario C	Nuevo flujo de clase AC_VO	97,9%
Escenario D	Nuevo flujo de clase AC_BE	92,8%

hicieron pruebas con dos umbrales de decisión ( $maxDelay1=30$  ms y  $maxDelay2=100$  ms) y en ambos se obtuvo 97,9% de precisión.

Es importante aclarar que se ha presentado una metodología, que en base a medidas y a técnicas de aprendizaje estadístico, permite realizar de decisiones de control de admisión. Para utilizar la metodología para clasificar alcanza con mandar los paquetes de medida con un tiempo entre paquetes de 10 ms para obtener excelentes resultados. Esto es importante porque genera una sobrecarga en la red que no afecta a los usuarios presentes al momento de realizar el monitoreo.

## 6.4. Herramientas para decisiones de Control de Admisión para una red 802.11e

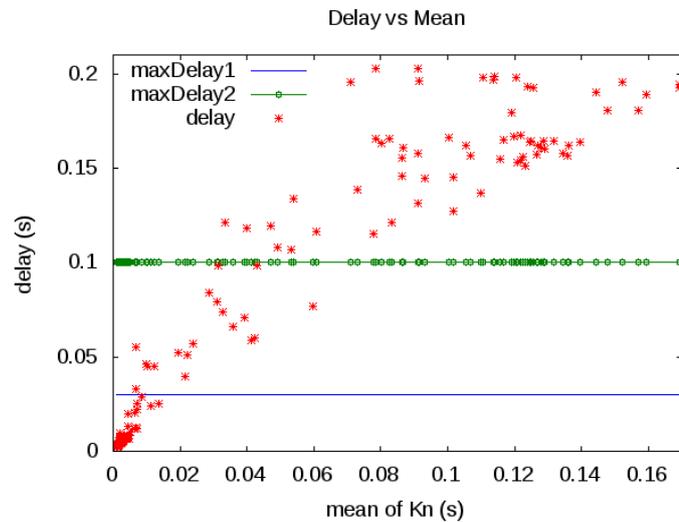


Figura 6.14: Delay obtenido para la nueva conexión de clase 1 vs Media de  $K_n$ . Enlace punto a punto 802.11e, escenario C

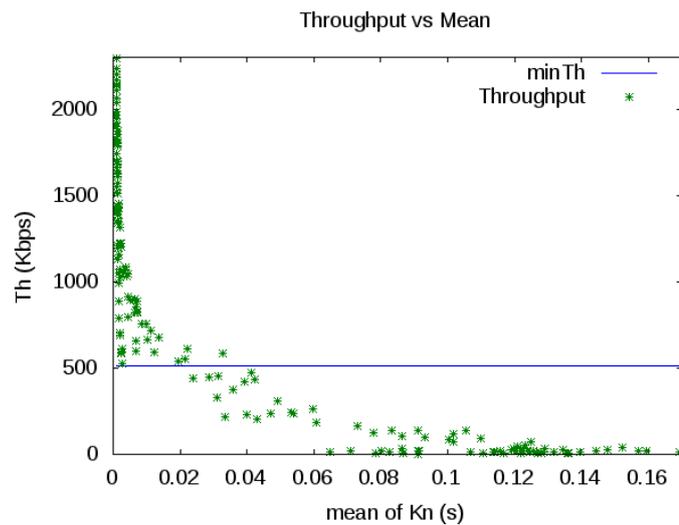


Figura 6.15: Throughput obtenido para la nueva conexión de clase 2 vs Media de  $K_n$ . Enlace punto a punto 802.11e, escenario D.

### 6.4.4. Decisiones de Control de Admisión considerando no afectar a los flujos presentes en la red con el ingreso de la nueva conexión

La idea en este caso es probar la validez de la metodología cuando el criterio de clasificación no se basa únicamente en las características de la nueva conexión, sino que también se debe garantizar que (con el ingreso de la nueva conexión en la red) los demás flujos no se vean afectados según sus requerimientos de QoS.

## Capítulo 6. Simulaciones

Ya vimos que existe una relación entre los parámetros de la nueva conexión y el estado de la red estimado por las estadísticas de los tiempos entre arribos y partidas de los paquetes de medida. Ahora, el problema se traduce en evaluar si es posible obtener alguna relación entre el estimador del estado de la red (algún estadístico de  $K_n$ ) y los parámetros de QoS de la conexión más comprometida al ingreso de la nueva. Vamos a considerar **conexión más comprometida** a aquella que tiene el menor throughput o tiene el mayor delay (según corresponda) una vez ingresada la nueva conexión en la red.

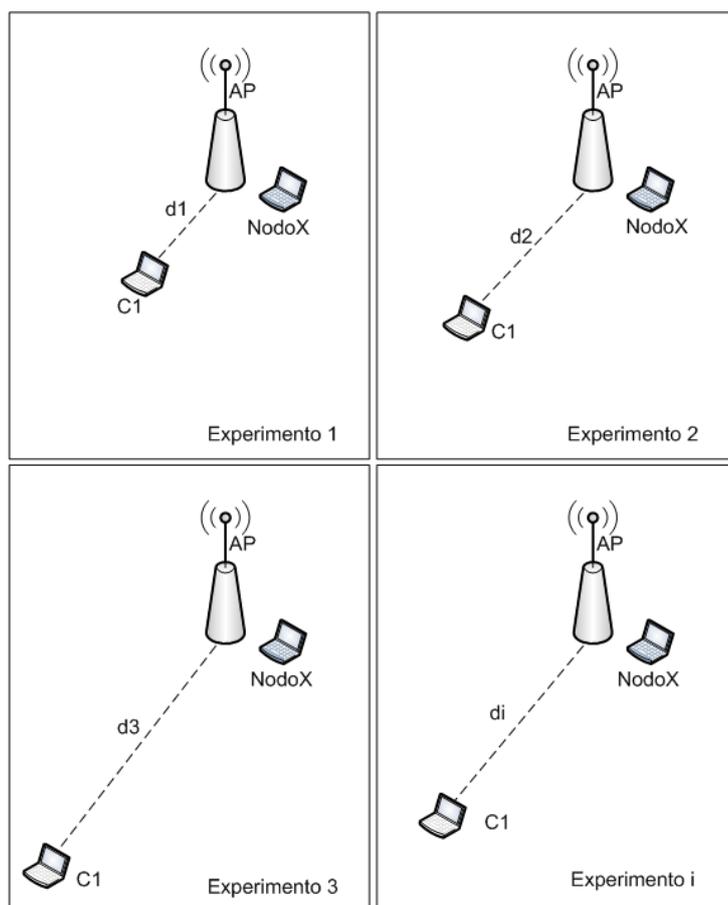


Figura 6.16: Topología de red donde lo único que varía entre simulaciones es la posición del nodo C1.

Un primer paso es analizar una topología bien simple: un AP, un nodo cliente (C1) y el nodo NodoX (este último será el encargado de generar la nueva conexión). Se realizaron varias pruebas variando la distancia entre el nodo C1 y el AP siendo la conexión entre éstos la más comprometida (Ver figura 6.16). Los paquetes de medida, como se ha detallado a lo largo del documento, se envían desde el nodo nuevo que quiere acceder a la red (NodoX en este caso).

#### 6.4. Herramientas para decisiones de Control de Admisión para una red 802.11e

En la gráfica 6.17 se muestra la relación existente entre el throughput obtenido en la conexión C1-AP y la media de  $K_n$ . Según se observa, parece razonable pensar que se puede lograr un criterio clasificatorio considerando los requerimientos de QoS de las conexiones existentes en la red, únicamente con el envío de los paquetes de medida y conociendo la cantidad de conexiones presentes. Una forma de explicar esto, es que al aumentar la distancia entre los nodos C1 y AP, uno de los fenómenos que se da es el aumento de la tasa de retransmisiones. Otro de los fenómenos, en el caso de tener encendido el multirate, es que la tasa de capacidad física de la conexión C1-AP disminuye al aumentar la distancia entre estos nodos. Ambas situaciones impactan en el desempeño de los paquetes de medida, y por lo tanto, parece lógico pensar en realizar decisiones considerando el desempeño de los flujos existentes.

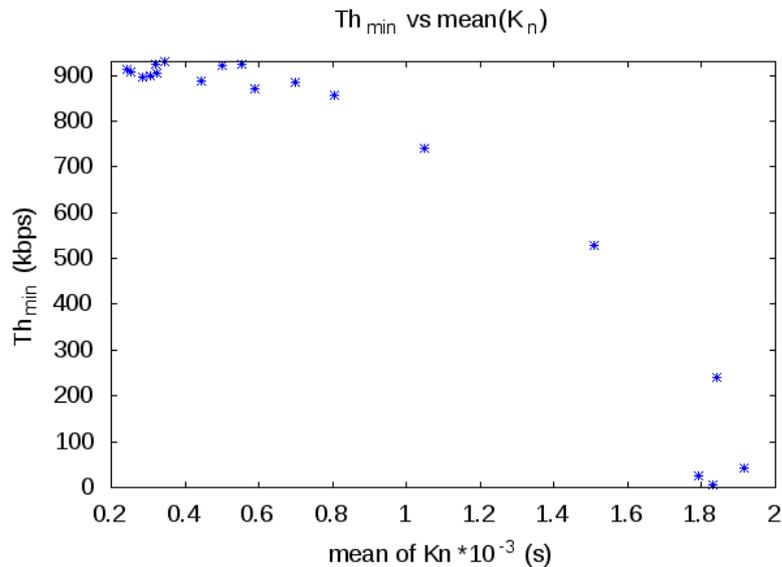


Figura 6.17: Throughput de la conexión más comprometida (C1-AP) en función de la media de  $K_n$  al variar la distancia entre el AP y C1. El NodoX (que es donde se origina la nueva conexión) se mantiene en la misma posición en todas las simulaciones.

El procedimiento en este caso sería el siguiente: para entrenar el sistema, una vez habilitada la nueva conexión se determina cuál es la más comprometida (puede ser la nueva u otra de las ya existentes). Luego se envía el tren de paquetes de medida para determinar el estado de la red. Por cada experimento se construye  $Z$  en función de si la conexión más comprometida cumple con los requisitos de QoS esperados y a su vez se registra la cantidad de conexiones existentes. En este caso el modelo se puede expresar como:  $Z = f(X, N)$ , siendo  $N$  la cantidad de conexiones,  $X$  el estado de la red y  $Z$  una variable binaria. Una vez entrenado, el modelo debe ser verificado.

## Capítulo 6. Simulaciones

Como prueba de validación del procedimiento mencionado se utilizó la Topología I definida anteriormente (WLAN 802.11e donde todos los nodos se encuentran a una distancia promedio de 30 metros, ver figura 6.8). Se realizó una clasificación considerando el siguiente criterio: una nueva conexión de clase 2 será aceptada en la red siempre que se le pueda garantizar un mínimo throughput dadas ciertas condiciones de stress de la red y además los flujos existentes no se vean afectados con el ingreso de la nueva conexión (los de clase 1 tengan un delay menor que cierto umbral y los de clase 2 un throughput mayor a cierto límite). En el caso de esta prueba, la topología es fija, en cada experimento sólo se cambió la cantidad de conexiones de cada clase presentes en la red.

Luego de utilizar SVM-C se obtuvo una precisión del 84 % (donde los paquetes de medida se enviaron con un tiempo entre partidas de 5 ms). Esto permite afirmar que la metodología puede ser utilizada, en estos contextos, como lo más cercano a lo que conocemos como un mecanismo de control de admisión propiamente dicho. Este procedimiento se verificó para las topologías explicadas antes, queda para trabajo futuro ver como implementar este procedimiento para topologías mas complejas.

# Capítulo 7

## Validación Experimental considerando 802.11e y WiMAX

### 7.1. Introducción

La metodología propuesta fue testeada de forma experimental en dos topologías simples pero típicas en escenarios rurales aislados. Las redes se montaron en un laboratorio de la Universidad del Rey Juan Carlos (Madrid- España). En la figura 7.1 se muestra uno de los escenarios instalados que consistió en una red híbrida formada por dos segmentos, uno WiMAX y otro WiFi. La otra arquitectura testeada consistió en un único salto WiFi tal como se esquematiza en la Figura 7.2. En ambos escenarios el segmento WiFi se implementó basado en el estándar IEEE 802.11e.

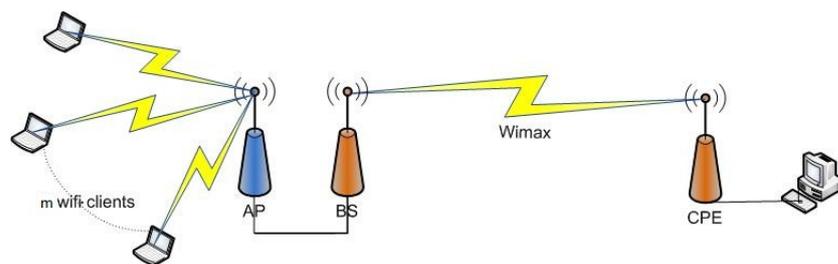


Figura 7.1: Topología A: Red Híbrida WiMAX ( $f_1=5$  GHz) y WiFi ( $f_2=2,4$  GHz).

En este capítulo se detallará el equipamiento y herramientas utilizadas para el armado de las maquetas. Se enumerarán las limitaciones presentadas y finalmente se presentarán los resultados obtenidos.

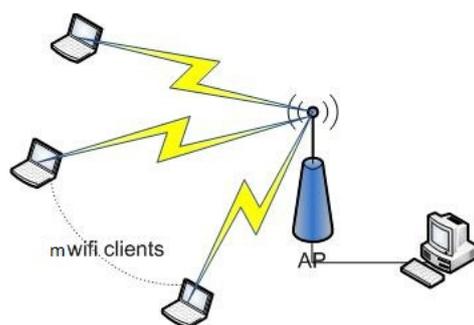


Figura 7.2: Topología B: Red de Área Local Inalámbrica ( $f=2,4$  GHz).

## 7.2. Descripción del escenario de trabajo

A continuación se explicarán en detalle los equipos que intervinieron en las pruebas así como también las configuraciones y validaciones básicas realizadas.

### 7.2.1. Equipamiento WiMAX

Se utilizaron equipos de Alentia Systems [66], en particular se usó una estación base y una estación suscriptor con las siguientes características:

Tabla 7.1: Descripción Equipos WiMAX.

Equipo WiMAX	Descripción
Estación Base (BS)	Modelo: ARBA-556 Firmware: 3.3.7133 Banda de Operación: 5GHz Antena: Sectorial
Estación Suscriptor (SS)	Modelo: CPE-56-PROA Antena: Sectorial

*Observación:* En un principio se tuvo como objetivo verificar la metodología en una topología más compleja: considerando dos saltos WiMAX. Una vez montados los equipos necesarios (la BS y los dos CPE), se constató un mal desempeño de la red. Luego de realizar diversas pruebas de validación de ancho de banda y disponibilidad de los enlaces, se concluyó que el mal desempeño era debido a la cercanía de los equipos por encontrarse dentro de un laboratorio (esto generaba saturación entre ellos). Se realizaron varios intentos para lograr estabilizar la red

## 7.2. Descripción del escenario de trabajo

(ej.: bajando la potencia de transmisión) pero no se consiguió un buen resultado. Por tal motivo se optó por simplificar la topología a un sólo salto WiMAX.



Figura 7.3: Equipo WiMAX usado.

### 7.2.2. Equipamiento WiFi

En este caso se utilizaron placas ALIX ([68], [67]) y laptops. En la tabla 7.2 se describen las características particulares de dichos equipos. Tal como se observa en las figuras 7.1 y 7.2, en ambas topologías hay un nodo que actúa de Access Point (AP: punto de acceso de los clientes). Para el AP se utilizó una de las placas ALIX, por otro lado, los demás nodos WiFi (clientes) fueron representados por laptops y por ALIXs.

Tabla 7.2: Descripción Equipos WiFi.

Equipo WiFi	Descripción
Placa ALIX	SO: Linux Voyage v0.6.5, kernel: 2.6.30 [69] Driver: MadWiFi (soporte de 802.11e) Radio: Mikrotik R52H Chipset: Atheros
Laptop	SO: Linux Ubuntu 11.10

*Observación 1:* Las placas ALIX permiten conectar dos radios independientes (cuentan con dos slots miniPCI) por lo que se pueden emular dos clientes WiFi

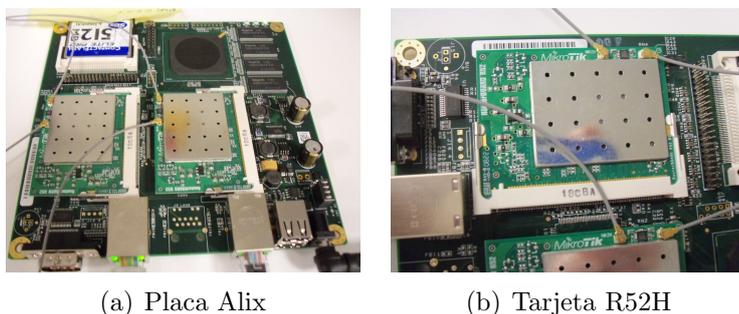


Figura 7.4: Hardware usado para los Nodos WiFi

distintos por ALIX. Entre laptops y ALIX se lograron  $m = 14$  clientes (cantidad suficiente para introducir carga en la red necesaria para el testeo de la metodología).

*Observación 2:* En la topología A, la conexión entre el AP y la BS se realizó utilizando un cable de red (ethernet).

### 7.2.3. Equipamiento para Servidores

En ambas topologías se utilizó una PC fija conectada vía ethernet a un equipo WiMAX (CPE) o WiFi (AP) según corresponda como se muestra en las figuras representativas (7.1 y 7.2). En el resto del documento a esta PC se la denominará *Servidor*. Además se utilizó otra PC que se conectó a todos los equipos a través de la red de gestión. Ésta se denominará PC\_Log ya que se utilizó como la encargada de recolectar y procesar los logs. También desde ésta se lanzaron las pruebas enviándole distintas órdenes a cada cliente. A ambas PC se les instaló Linux Ubuntu 11.10 (al igual que a las laptops).

### 7.2.4. Sincronización de Relojes

Para realizar medidas de retardo y jitter es necesario tener los equipos (clientes y servidor) sincronizados. Sino se realiza una buena sincronización dichas medidas carecen de sentido (ejemplo: delays negativos).

Para lograr esto se montó una red cableada adicional (red de gestión) entre todos los equipos que intervinieron en las pruebas. La PC\_Log se configuró como servidor mediante *ntp* y los demás equipos se sincronizaban automáticamente y periódicamente usando la herramienta *ntpdate*<sup>1</sup>.

---

<sup>1</sup>Cada 1 segundo, en todos los equipos se ejecutaba *ntpdate* contra PC\_Log para no perder la sincronización.

## 7.2. Descripción del escenario de trabajo

Se alcanzaron niveles de offset del orden de la decena del milisegundo (pero en algunos casos fue del orden del milisegundo). Este punto resultará determinante a la hora de evaluar las medidas obtenidas de delay y jitter.

*Observación:* La sincronización resultó uno de los puntos más problemáticos a la hora de la puesta a punto de la maqueta para las pruebas. Lograr los niveles mencionados llevó varios días de pruebas e investigación del funcionamiento de ntp y ntpdate.

### 7.2.5. Elección de Frecuencias y Rate Capa Física

Los experimentos no se realizaron en un ambiente totalmente controlado. Se realizaron en los laboratorios de la Universidad por lo que se tuvo que lidiar con interferencia externa a pesar de elegir los canales “más libres”. La banda de 2,4 GHz se encontraba prácticamente toda ocupada, es por ello que se tomó la decisión de dejar sólo un tramo WiFi en las topologías a probar (si se configuraban dos saltos WiFi en 2,4 con los canales suficientemente separados para evitar la interferencia co-canal, uno de ellos siempre quedaba comprometido con la interferencia exterior afectando las medidas de los experimentos).

Por otro lado, todas las interfaces WiFi se configuraron para usar un rate de 11 Mbps. Tanto para la topología A como para la topología B, se realizó una validación del ancho de banda total de extremo a extremo al configurar el *Rate Físico* mencionado. Se realizaron varias medidas obteniendo (para ambos casos) más de 9 Mbps de ancho de banda efectivo ( $BW_e$ ) coincidiendo con lo teóricamente esperado. Las medidas de  $BW_e$  se realizaron tanto con IPERF [72] como con D-ITG [70] utilizando tráfico UDP, y en ambas herramientas se consiguieron resultados similares.

### 7.2.6. Inyector de Tráfico: D-ITG

D-ITG (Distributed Internet Traffic Generator) fue el elegido para la generación de tráfico [70] [71]. Permite inyectar tráfico con patrones lo más realistas posibles y tiene un bloque “manager” que permite enviar órdenes a fuentes y sumideros de tráfico de forma remota. Una funcionalidad a resaltar de la herramienta es que permite marcar el tráfico y así obtener flujos de distintas clases de acceso. Tiene además incorporado un módulo para el tratamiento de logs y estadísticas. D-ITG se puede instalar en linux (así como en otro SO). En el caso particular de este trabajo se usó la versión 2.8.0-rc1.

*Observación:* Otro generador de tráfico que se evaluó fue IPERF [72] pero resultó limitado, en comparación con D-ITG, para las medidas que se deseaban realizar.

## Capítulo 7. Validación Experimental considerando 802.11e y WiMAX

### Generación de tráfico

Tal como se explicó en capítulos anteriores, la idea es estudiar la red inalámbrica cuando existen dos clases de flujos. Para las pruebas se usaron las siguientes AC: AC\_VI y AC\_BE (definidas en EDCA) por lo que el D-ITG debe marcar los los paquetes usando el campo DSCP (Differentiated Services Code Point) de acuerdo a la tabla 7.3.

Tabla 7.3: Mapeo entre Categorías de Acceso (AC) y campo DSCP (Driver MadWifi)

DS field	DSCP	EDCA AC
0x00	0x00	AC_BE
0xa0	0x28	AC_VI

A continuación se muestran dos ejemplos que ayudarán a comprender el mapeo. Se muestra la generación de dos flujos “iguales” (mismo protocolo de transporte, misma duración, mismo tamaño de paquete, mismo tiempo entre paquetes, mismo origen y destino) pero uno está marcado y el otro no:

```
-a ip_dest -rp port_dest -b 0x00 -C 80 -c 1492 -t 60000  
-a ip_dest -rp port_dest -b 0xa0 -C 80 -c 1492 -t 60000
```

### 7.2.7. Configuraciones Básicas

Los equipos WiMAX se configuraron a través de la interfaz web que provee el fabricante.

Para los nodos WiFi y PCs se realizaron scripts (shell) para que, al iniciar cargaran su configuración automáticamente:

- Configuración de IPs (para la red de gestión y para la red inalámbrica).
- Configuración de rutas
- Habilitación de WMM (EDCA) para el caso de los equipos WiFi
- Rate capa física en 11 Mbps (en los equipos WiFi)
- Configuración y habilitación de la sincronización periódica vía ntpdate

## 7.3. Configuración de QoS y Verificación

### 7.3.1. Configuración y verificación del segmento EDCA

Se dejó incambiada la configuración EDCA que viene por defecto en el driver utilizado. En la siguiente tabla se muestran los parámetros de acceso al medio para

### 7.3. Configuración de QoS y Verificación

las dos clases de acceso elegidas para las pruebas.

Tabla 7.4: Parámetros EDCA usados.

AC	CWmin	CWmax	AIFSN	TXOPLimit
AC_VI	3	4	2	0.003008
AC_BE	3	10	2	0

Se verificó el correcto funcionamiento del marcado de los paquetes realizando capturas de tráfico con *tcpdump* [73] en los extremos de la red.

También se verificó el correcto desempeño de EDCA al usarlo como mecanismo de acceso al medio. Para esto, desde cierto nodo WiFi origen se enviaron dos flujos simultáneos durante el mismo tiempo hacia un mismo nodo destino. Ambos flujos se construyeron iguales salvo por la marca que los clasificaba a cada uno en una clase de acceso mencionadas (los flujos en conjunto saturaban la capacidad máxima del enlace). Los resultados obtenidos de esta prueba se pueden observar en las gráficas de la figura 7.5. Claramente se ve como se reparte de manera desigual el tiempo de acceso al medio, ocasionando que el throughput obtenido sea mayor y el retardo mucho menor para el flujo de clase AC\_VI comparado con el de clase AC\_BE.

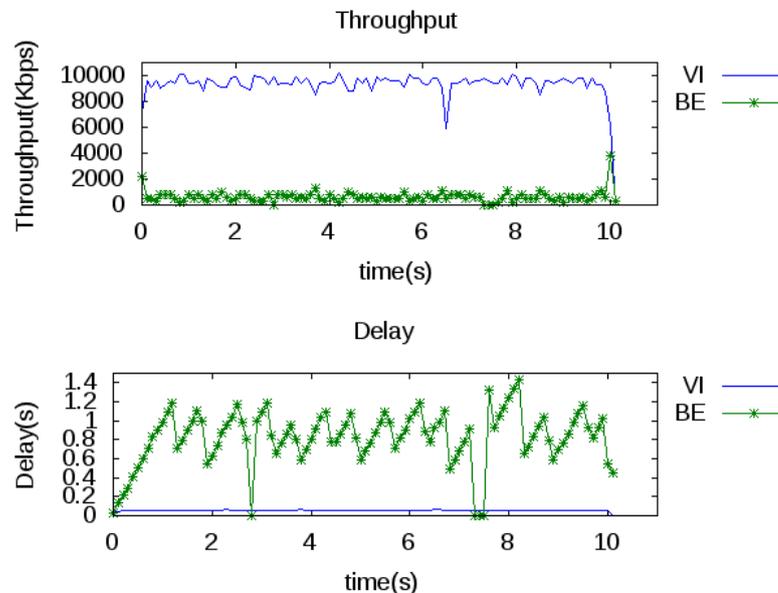


Figura 7.5: Verificación del funcionamiento de EDCA.

### 7.3.2. Configuración y verificación del segmento WiMAX

Una vez verificado el desempeño esperado en el tramo WiFi-EDCA se procedió a validar el tramo WiMAX (necesario para la Topología A).

WiMAX permite brindar QoS parametrizada, para ello es necesario configurar distintos flujos de servicio (tanto para tráfico downlink como para uplink). La configuración se realiza sólo en la BS. Para el caso particular de las pruebas, donde se consideraron sólo dos tipos de tráfico, se configuraron dos pares (uplink/downlink) de flujos de servicio (SF): rtPS y BE.

En la Figura 7.6 se muestran los dos pares de SF configurados en los que se puede observar las prioridades asignadas (QoS Priority<sup>2</sup>) y el ancho de banda garantizado para cada uno. Además de los parámetros mencionados, para cada SF es posible configurar otros umbrales (ej.: Maximum Jitter, Maximum Latency), para éstos se tomaron como base las recomendaciones de ITU [74] según los tipos de tráfico que se intentaron representar en las pruebas.

**Provisioned Flows (4)**

**DL Services**

Description (ID)	QoS Summary	ARQ	Action
BE_DL(1)	Prio: 4 - Max: 10000Kbps - Min: 0Kbps	no	Delete
VI-DL(3)	Prio: 1 - Max: 6000Kbps - Min: 6000Kbps	no	Delete

**UL Services**

Description (ID)	QoS Summary	ARQ	Action
BE_UL(2)	BE - Prio: 4 - Max: 10000Kbps - Min: 0Kbps	no	Delete
VI-UL(4)	RTPS - Prio: 1 - Max: 6000Kbps - Min: 6000Kbps	no	Delete

Add provisioned flow

Figura 7.6: Configuración de los Flujos de Servicio - WiMAX

Además de la definición de SF se necesita definir cómo se clasificarán los paquetes para matchear con alguno de los SF mencionados. Esto se realizó mediante el campo DSCP de la cabecera IP. Ver tabla 7.5.

Tabla 7.5: Mapeo entre Categorías de Acceso de EDCA, Flujos de Servicio de WiMAX y campo DSCP.

DS field	DSCP	EDCA AC	WiMAX SF
0x00	0x00	AC_BE	BE
0xa0	0x28	AC_VI	rtPS

<sup>2</sup>Puede tomar los valores entre 0 y 7, cuanto mayor sea, mayor es su prioridad

## 7.4. Automatización de las pruebas y generalidades

En el cuadro 7.6 se demuestran algunas pruebas realizadas y resultados obtenidos para verificar las configuraciones mencionadas.

Tabla 7.6: Validación de la configuración del segmento WiMAX.

Prueba	Flujos inyectados	Bitrate(Kbit/s)
1	<code>./ITGSend -a 10.10.1.1 -rp 9201 -b 0xa0 -T UDP -C 1500 -c 1000 -t 10000</code>	5580,83
	<code>./ITGSend -a 10.10.1.1 -rp 9202 -b 0x00 -T UDP -C 1500 -c 1000 -t 10000</code>	3721,33
2	<code>./ITGSend -a 10.10.1.1 -rp 9202 -b 0x00 -T UDP -C 1500 -c 1000 -t 10000</code>	9097,19
3	<code>./ITGSend -a 10.10.1.1 -rp 9201 -b 0xa0 -T UDP -C 1500 -c 1000 -t 10000</code>	5695,24

Una vez verificado el correcto funcionamiento de los dos segmentos por separado se validaron ambas topologías de extremo a extremo obteniendo el comportamiento deseado.

## 7.4. Automatización de las pruebas y generalidades

Como se define en la metodología y tal como se verificó en las simulaciones, se debe lograr obtener un número importante de medidas para poder entrenar el sistema y lograr clasificar con poco error. Para esto se trabajó en la realización de scripts en bash que se encargaban de lanzar cada experimento y almacenar los datos correspondientes en la PC\_Log.

Cada experimento consistió en tres fases (cada una de 60 seg de duración):

1. Fase 1: Se envía la orden de generar tráfico a distintos clientes (la cantidad de flujos de cada tipo varía entre experimentos).
2. Fase 2: Se realiza el mismo envío de flujos de la fase 1, pero ésta vez se suma la “nueva conexión” que se desea clasificar.
3. Fase 3: Se realiza el mismo envío de flujos de la fase 1, pero desde el nodo originario de la nueva conexión se envía el tren de paquetes de medida.

Las fases emulan lo realizado para las simulaciones en particular para lo detallado en la sección 6.4. En la primer fase se tiene una foto del estado de la red sin la nueva conexión (se pueden determinar los parámetros obtenidos por los flujos presentes en la red), en la segunda se obtiene el estado de la red una vez que se aceptó la nueva conexión (se puede conocer el desempeño de la nueva conexión), y finalmente, se realiza el envío de paquetes de medida para lograr estimar la utilización de aire en la red previo a que ingrese la nueva conexión.

## Capítulo 7. Validación Experimental considerando 802.11e y WiMAX

Al igual que en las simulaciones, el estimador del estado del canal ( $X$ ), se logra con los tiempos de arribo y partida de los paquetes de medida (obtenidos de la fase 3). En el Apéndice A se detallan los scripts más importantes utilizados para automatizar el procedimiento.

Para lograr esta automatización se insumió mucho tiempo ya que se tuvieron que lidiar con “bugs” del D-ITG y limitantes de los equipos y software usados.

*Observación:* Todos los flujos cursados en la red correspondientes a la clase AC\_BE se configuraron como flujos TCP a tasa constante, y los demás se configuraron UDP con tasa de arribos exponencial. Por más detalles ver Apéndice A.

### 7.5. Realización de Medidas

A continuación se describe el método utilizado para obtener cada medida.

#### Medida de throughput

A partir de los logs recolectados con D-ITG es posible obtener el throughput instantáneo durante los 60 seg que dura cada inyección de tráfico. Dado que es técnicamente imposible lograr que todos los nodos comiencen el envío de paquetes exactamente en el mismo instante de tiempo, para el cálculo de throughput se tomaron los valores del instantáneo entre 10 y 50 seg. Se usará el valor medio para las decisiones de clasificación.

D-ITG permite generar archivos que pueden ser interpretados por octave [75] (software utilizado para la realización de cálculos y gráficos).

#### Medida de delay

Se siguió el mismo método que se explicó para el throughput.

#### Medida de jitter

Debido a los niveles de sincronización alcanzados y a la cercanía de los equipos (por tratarse de una maqueta montada dentro de un laboratorio), las medidas obtenidas de jitter se consideran poco fiables. No es posible tomar decisiones con ellas.

#### Medida del Estado de la Red $X$

Usando la módulos disponibles de D-ITG es posible obtener un archivo con los tiempos de partida y llegada de cada paquete de medida. El archivo resultante

puede ser procesado con octave. También se toman los datos entre 10 y 50 seg (para tener el sistema en “estado estacionario”).

## 7.6. Análisis de Resultados

En esta sección se resumirán los resultados obtenidos al aplicar la metodología propuesta en las topologías A y B detalladas al comienzo del capítulo. Para construir el estado de la red ( $X$ ) se envió la ráfaga de paquetes de medida en cada experimento (fase 3).

Características del tren de paquetes inyectado:

- tamaño de paquetes: 10 bytes (payload)
- tiempo entre paquetes: 10 ms
- duración de la ráfaga: 60 seg
- protocolo de clase de transporte: UDP
- clase de tráfico: AC\_VI (mayor prioridad)

Dados los datos anteriores esto supone una sobrecarga en la red de aproximadamente 0,1 % (considerando que la capacidad máxima es alrededor 9 Mbps). Lo anterior permite afirmar que la metodología no resulta invasiva.

Se realizaron distintas pruebas para las dos topologías y para distintos tipos de flujo asociados a la nueva conexión.

Se desea validar la metodología para el caso de aplicación en decisiones de control de admisión. El objetivo es lograr decidir cuándo es posible aceptar en la red la nueva conexión y cuándo no. La decisión se realizó siguiendo el criterio mencionado en el la sección 4.3. Cuando la conexión a clasificar es de clase AC\_VI se aceptará si se le garantiza un retardo menor a cierto umbral, si es del tipo AC\_BE se aceptará si se le garantiza un throughput mayor a cierto límite. Considerando dicho criterio, midiendo en la fase 2 de cada experimento el delay o el throughput obtenido por la nueva conexión (según corresponda), se generó la secuencia binaria ( $Z$ ) que resulta de comprar la medida con el umbral correspondiente ( $maxDelay$  o  $minTh$ ).

Una vez obtenido un número importante de medidas, se aplicó SVM-C y se obtuvieron los resultados mostrados en el cuadro 7.7. Se lograron buenos niveles de precisión a pesar de lidiar con varios problemas, entre ellos el más importante: interferencia externa variable.

Tal como se mencionó, los niveles de sincronización obtenidos introducen errores significativos en las medidas de delay (delays obtenidos muy pequeños del

## Capítulo 7. Validación Experimental considerando 802.11e y WiMAX

Tabla 7.7: Resumen de resultados obtenidos al aplicar SVM-C usando la media de  $K(n)$  como estimador del estado de la red.

Escenario	Descripción	Precisión obtenida
Escenario 1	Topología A, <i>nuevo flujo</i> de clase AC_BE	85 %
Escenario 2	Topología B, <i>nuevo flujo</i> de clase AC_BE	88 %
Escenario 3	Topología B, <i>nuevo flujo</i> de clase AC_VI	78 %

orden del milisegundo). Esto explica que la estimación para el caso 3 sea la peor de las tres.

No se considera, al problema de sincronización, una limitante de la técnica propuesta ya que en escenarios reales, los equipos en general están a mayor distancia (especialmente en redes rurales) donde los retardos percibidos son mucho mayores.

Se intentaron realizar las pruebas nuevamente considerando tráfico de ida y vuelta (o sea, sin errores de sincronización). No se logró completar este experimento debido a que se encontró un bug en la versión usada de D-ITG que impactaba en la generación y medición del tráfico de ida y vuelta. Se probaron versiones anteriores pero existía otra serie de bugs que imposibilitaron las pruebas.

En la figura 7.7 se muestra la relación entre el throughput obtenido por la conexión a clasificar (perteneciente a clase AC\_BE) junto con la media de  $K_n$  para el Escenario 1. Resulta interesante resaltar que el comportamiento de la curva de la Figura 7.7 es similar a lo observado en 6.12 y 6.15 a pesar de que se trata de distintas topologías de red.

Observación: Utilizando otras medidas estadísticas para  $X$  no se mejoran los resultados obtenidos al usar sólo la media.

Una contribución de este trabajo es que la técnica no sólo es verificada en simulaciones, sino que es testeada también en la práctica, dando una mayor confiabilidad.

## 7.6. Análisis de Resultados

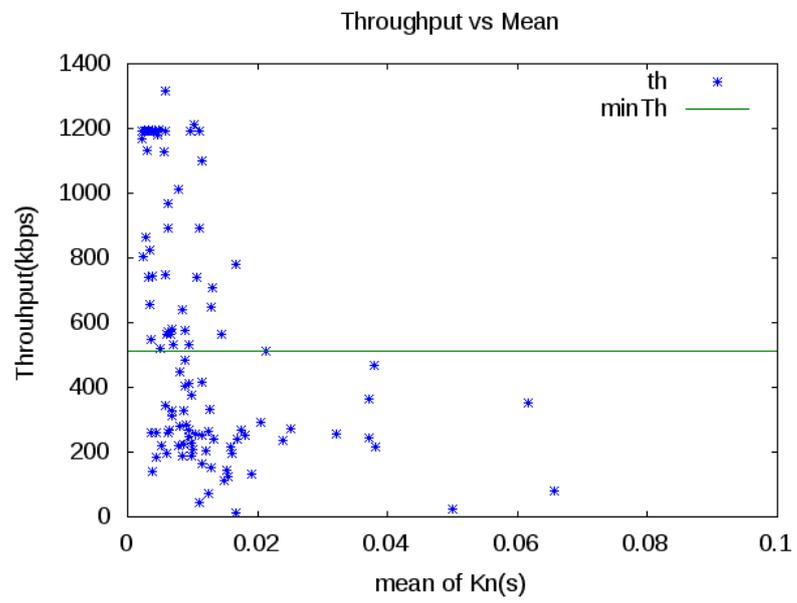


Figura 7.7: Throughput obtenido por la nueva conexión vs Media de  $K_n$  (Escenario 1)

Esta página ha sido intencionalmente dejada en blanco.

Parte IV

CONCLUSIONES Y TRABAJO  
FUTURO



# Capítulo 8

## Conclusiones y Trabajo Futuro

### 8.1. Conclusiones

- La principal contribución de este trabajo es la metodología propuesta que permite, a partir de mediciones en la red y de técnicas de aprendizaje estadístico, realizar estimaciones de ciertos parámetros de QoS y tomar decisiones de control de admisión en redes inalámbricas.
- Se presentaron resultados obtenidos a partir de simulaciones y de pruebas experimentales y en ambos casos se obtuvieron muy buenas precisiones (tanto para la estimación de parámetros de QoS, así como también a la hora de realizar clasificación decidiendo cuál conexión puede acceder a la red y cuál no).
- A pesar de que la metodología implica la realización de mediciones activas en la red, se demostró que la misma no resulta intrusiva una vez entrenado el sistema.
- La técnica propuesta es independiente de la tecnología de la red inalámbrica y puede ser usada en redes híbridas y mutisalto. En particular, se mostró el desempeño de la misma en una red WiFi-WiMax.

### 8.2. Trabajo Futuro

Este trabajo podría ser el puntapié para otras investigaciones. A continuación se describen algunas posibles líneas a futuro:

En primer lugar, extender la metodología para confeccionar un mecanismo de control de admisión propiamente dicho. Ver como identificar cuando se trata de una nueva conexión para evaluar su aceptación (o no) en la red. La decisión debe realizarse considerando garantías de cierto servicio a la nueva conexión (según el tipo de aplicación) y procurando además, que los demás flujos presentes en la red no se vean afectados con la entrada de la nueva conexión. En este tema se dieron

## Capítulo 8. Conclusiones y Trabajo Futuro

unos primeros pasos, pero no se llegó a elaborar el algoritmo en su totalidad.

En segundo lugar, resultaría interesante realizar una verificación exhaustiva de la técnica en topología más complejas y en enlaces de larga distancia (de particular interés en las redes rurales).

Otra línea de trabajo podría ser, continuar investigando qué otras medidas estadísticas de  $K_n$  pueden mejorar la estimación y clasificación obtenida, o evaluar de que otras maneras se puede estimar la utilización del aire de forma más precisa (sin ser más invasivo).

En la realización de decisiones de control de admisión, por simplificación, se tomó un único parámetro de decisión por clase de tráfico (delay o throughput). Es importante resaltar que una solución completa debería considerar simultáneamente requerimientos de throughput, delay y pérdidas de paquetes para cualquier flujo en tiempo real. Una línea de trabajo futuro, sería probar la metodología con criterios de decisión más completos.

Parte V  
APÉNDICES



# Apéndice A

## Scripts utilizados en Pruebas de Laboratorio

A continuación se muestran los principales scripts realizados para las pruebas de laboratorio. Todos los scripts se ejecutan en PC\_Log, a través la red de gestión (10.10.10.x/24) se envían las órdenes a los distintos equipos.

### Script generador de tráfico - gen\_flujos.sh

Parámetros de entrada:

- identificador de la prueba *Num\_prue*
- cantidad de flujos de clase BE con origen en los clientes wifi y destino el servidor - *m*
- cantidad de flujos de clase VI con origen en los clientes wifi y destino el servidor - *n*
- cantidad de flujos de clase BE con origen en el servidor y destino los clientes wifi - *s*
- cantidad de flujos de clase VI con origen en el servidor y destino los clientes wifi - *r*

Función: Se encarga de conformar y ejecutar un script de nombre *Num\_prue*. Dicho script se forma tomando tantas líneas de los archivos lista\_m.txt, lista\_n.txt, lista\_r.txt y lista\_s.txt como se indica en las variables *m*, *n*, *r* y *s* respectivamente. Al ejecutarse *Num\_prue* se realizan órdenes remotas a través de la herramienta ITGManager para que los distintos clientes comiencen los envíos de tráfico que correspondan.

## Apéndice A. Scripts utilizados en Pruebas de Laboratorio

```
#!/bin/bash
echo "PARTE A "
echo " "
echo "TIPOS DE TRAFICO : "
echo "======"
echo " "
echo "M = CLASE 1 TCP CONSTANTE (clientes hacia el servidor)"
echo "N = CLASE 0 TIPO I UDP Exponencial (clientes hacia el servidor)"
echo "R = CLASE 0 TIPO I UDP Exponencial (servidor hacia los clientes)"
echo "S = CLASE 1 TCP CONSTANTE (servidor hacia los clientes)"

echo " "
echo CLASES ... Lanzadas
echo "Numero de prueba: $1"
Num_prue=$1
echo "Numero de clases M: $2"
Num_M=$2
echo "Numero de clases N: $3"
Num_N=$3
echo "Numero de clases R: $4"
Num_R=$4
echo "Numero de clases S: $5"
Num_S=$5
echo " "

echo "# FLUJOS M = $Num_M " > $Num_prue
head -$Num_M $HOME/script/lista_m.txt > $Num_prue
echo "# FLUJOS N = $Num_N" >> $Num_prue
head -$Num_N $HOME/script/lista_n.txt >> $Num_prue
echo "# FLUJOS R = $Num_R" >> $Num_prue
head -$Num_R $HOME/script/lista_r.txt >> $Num_prue
echo "# FLUJOS S = $Num_S" >> $Num_prue
head -$Num_S $HOME/script/lista_s.txt >> $Num_prue
cd $HOME/DIGT/D-ITG-2.8.0-rc1/bin/
chmod +x $HOME/script/$Num_prue
bash $HOME/script/$Num_prue
```

Este script se invoca 3 veces desde el script principal:

- Estado inicial de la red (cantidad de flujos=  $m + n + r + s$ )
- Envío del flujo de aplicación a clasificar (cantidad de flujos= $m+n+r+s+1$ )
- Envío de paquetes de medida (cantidad de flujos= $m + n + r + s + 1$ )

## Ejemplos de archivos lista: lista\_r.txt y lista\_s.txt

lista\_r.txt

```
./ITGManager 10.10.10.1 -a 10.10.4.25 -rp 9101 -b 0xa0 -E 90 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.3.25 -rp 9102 -b 0xa0 -E 90 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.4.24 -rp 9103 -b 0xa0 -E 90 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.3.24 -rp 9104 -b 0xa0 -E 90 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.4.23 -rp 9105 -b 0xa0 -E 90 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.3.23 -rp 9106 -b 0xa0 -E 90 -c 1492 -t 60000 &
...
...
```

lista\_s.txt

```

./ITGManager 10.10.10.1 -a 10.10.3.11 -rp 9401 -b 0x00 -T TCP -C 100 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.4.12 -rp 9402 -b 0x00 -T TCP -C 100 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.3.13 -rp 9403 -b 0x00 -T TCP -C 100 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.4.14 -rp 9404 -b 0x00 -T TCP -C 100 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.3.22 -rp 9405 -b 0x00 -T TCP -C 100 -c 1492 -t 60000 &
./ITGManager 10.10.10.1 -a 10.10.4.22 -rp 9406 -b 0x00 -T TCP -C 100 -c 1492 -t 60000 &
...
...

```

## Script principal

Se generan tantas pruebas como combinaciones se tienen de las variables  $m$ ,  $n$ ,  $r$  y  $s$  que representan la cantidad de flujos de cada clase en cada sentido. A partir de cada prueba se emula un estado distinto de cada red que se determina con el envío del tren de paquetes.

Se utilizan otros dos scripts que no fueron mencionados: *inicializo.sh* y *moverLogs.sh* que se encargan de la inicialización de los demonios de ITGSend e ITGRecv en cada cliente y renombrar identificando los logs respectivamente.

```

#!/bin/bash

. ./inicializo.sh
sleep 60
prueba=1001

for (( r= 1; r<= 5; r=r+1 ))
do
    for (( s= 0; s<= 5; s=s+1 ))
    do
        for (( m= 0; m<= 6; m=m+1 ))
        do
            for ((n= 0; n<= 6; n++ ))
            do
                #red sola
                ./gen_flujos1.sh $prueba $m $n $r $s
                sleep 120

                # muevo los logs y reestablezco los demonios en los clientes
                ./moverLogs.sh $prueba A
                . ./inicializo.sh
                sleep 60

                #red con el nuevo flujo
                ./gen_flujos2.sh $prueba $m $n $r $s
                sleep 120

                # muevo los logs y reestablezco los demonios en los clientes
                ./moverLogs.sh $prueba B
                . ./inicializo.sh
                sleep 60

                #red con los paquetes de prueba
                ./gen_flujos3.sh $prueba $m $n $r $s
                sleep 120

                # muevo los logs y reestablezco los demonios en los clientes
                ./moverLogs.sh $prueba C
            done
        done
    done
done

```



# Apéndice B

## Scripts utilizados en las simulaciones de ns-2

Se muestra a continuación el ejemplo utilizado para simular una red compuesta por dos nodos wifi donde se cursan flujos de dos clases de servicio utilizando EDCA como mecanismo de acceso al medio. Este script se ejecuta en ns2.28 con el parche [59] que incluye todo lo referente a 802.11e.

```
# Un enlace punto a punto entre 2 nodos wifi

if {$argc == 0} {
    set val(run) 1
    set val(numFlowPON1) 2
    set val(numFlowP3N1) 0
    set val(numFlowPON0) 2
    set val(numFlowP3N0) 2
} else {
    set val(run) [lindex $argv 0]
    set val(numFlowPON1) [lindex $argv 1]
    set val(numFlowP3N1) [lindex $argv 2]
    set val(numFlowPON0) [lindex $argv 3]
    set val(numFlowP3N0) [lindex $argv 4]
}

# Link between nodes wireless
#Complete without delay box
set val(chan) Channel/WirelessChannel ;# Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11e ;# MAC type
set val(ifq) Queue/DTail/PriQ ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 100 ;# max packet in ifq
set val(cbrbrate) 10000000 ;# total traffic rate
set val(SimTime) 280 ;# Simulation Time
set val(startmin) 1 ;# starting time of each transmission
set val(startmax) 1.20 ;# idem startmin
set val(nn) 2 ;# number of mobilenodes,

# routing protocol
set val(rp) DSDV
```

## Apéndice B. Scripts utilizados en las simulaciones de ns-2

```
#size window
set val(x)      500
set val(y)      500

# Initialize Global Variables
set ns          [new Simulator]
set tracefd     [open enlace.tr w]
$ns trace-all $tracefd

global defaultRNG
set startrng [new RNG]
set rvstart [new RandomVariable/Uniform]
$rvstart set min_ $val(startmin)
$rvstart set max_ $val(startmax)
$rvstart use-rng $startrng
# for each simulation
for {set j 1} {$j < $val(run)} {incr j} {
    $defaultRNG next-substream
    $startrng next-substream
}

Mac/802_11e set PreambleLength_      72          ; # 144 large preamble - 72 short preamble
Mac/802_11e set PLCPHeaderLength_    24          ; # 48 large header - 24 short header
Mac/802_11e set dataRate_            11000000   ; # rate of wireless link
Mac/802_11e set basicRate_           2000000    ; # rate ACK
Mac/802_11e set RTSThreshold_        30000000   ; # Limit packet length for enabling RTS/CTS
Mac/802_11e set cfb_1                1          ; # Enable TXOP
Mac/802_11e set ShortRetryLimit_     7          ; # Maximum number of retransmission of RTS/ CTS
Mac/802_11e set LongRetryLimit_      7          ; # Maximum number of retransmission of ACK
Mac/802_11e set PLCPDataRate_        1.0e6      ; # rate PLCP (preamble+header)

# set up topography object
set topo        [new Topography]
$topo load_flatgrid $val(x) $val(y)
# Create God
set god_ [create-god [expr $val(nn)] ]
# Create channel
set chan_ [new $val(chan)]

# Create node(0) and node(1)
# configure node
$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace OFF \
    -macTrace OFF \
    -movementTrace OFF \
    -channel $chan_

# creo dos nodos
set n(0) [$ns node]
$n(0) set X_ 0
$n(0) set Y_ 0
$n(0) set Z_ 0
$n(0) random-motion 0
puts "node 0 created ..."
```

```

set n(1) [$ns node]
$n(1) set X_ 30
$n(1) set Y_ 0
$n(1) set Z_ 0
$n(1) random-motion 0
puts "node 1 created ..."

# probe packets
set udp_pack [new Agent/UDP]
$udp_pack set packetSize_ 10
$udp_pack set class_ 0
$udp_pack set prio_ 0
$ns attach-agent $n(1) $udp_pack
set null_pack [new Agent/Null]
$ns attach-agent $n(0) $null_pack
$ns connect $udp_pack $null_pack
puts "udp_pack agent created ..."

# node new1
#set udp_new1 [new Agent/UDP]
#$udp_new1 set packetSize_ 1500
#$udp_new1 set class_ 0
#$udp_new1 set prio_ 0
#$ns attach-agent $n(1) $udp_new1
#set null_new1 [new Agent/Null]
#$ns attach-agent $n(0) $null_new1
#$ns connect $udp_new1 $null_new1
#set cbr_new1 [new Application/Traffic/CBR]
#$cbr_new1 set rate_ 1000000
#$cbr_new1 set packetSize_ 1500
#$cbr_new1 attach-agent $udp_new1
#$ns at 100.00 "$cbr_new1 start"
#$ns at 200.00 "$cbr_new1 stop"
#puts "cbr_new1 node created ..."

# node new2
set tcp_new [new Agent/TCP/Newreno]
$tcp_new set packetSize_ 1500
$tcp_new set class_ 3
$tcp_new set prio_ 3
$ns attach-agent $n(1) $tcp_new
set sink_new [new Agent/TCPSink]
$ns attach-agent $n(0) $sink_new
$ns connect $tcp_new $sink_new
set ftp_new [new Application/FTP]
$ftp_new set packetSize_ 1500
$ftp_new attach-agent $tcp_new
$ns at 100 "$ftp_new start"
$ns at 200 "$ftp_new stop"
puts "tcp_new2 node created ..."

proc sendpacket {} {
    global ns udp_pack val
    set now [$ns now]
    $ns at [expr $now + 0.010] "sendpacket"
    set bytes [expr round (10)]
    $udp_pack send $bytes
}

# Traffic sources n1
# class 0
for {set i 1} {$i <= $val(numFlowPON1)} {incr i} {
    set null1($i) [new Agent/Null]

```

## Apéndice B. Scripts utilizados en las simulaciones de ns-2

```
    $ns attach-agent $n(0) $null1($i)
}
for {set i 1} {$i <= $val(numFlowPON1)} {incr i} {
    set udp1($i) [new Agent/UDP]
    $udp1($i) set packetSize_ 1500
    $udp1($i) set class_ 0
    $udp1($i) set prio_ 0
    $ns attach-agent $n(1) $udp1($i)
    puts "udp1($i) class 0 node created ..."
}
for {set i 1} {$i <= $val(numFlowPON1)} {incr i} {
    $ns connect $udp1($i) $null1($i)
}
for {set i 1} {$i <= $val(numFlowPON1)} {incr i} {
    set exp1($i) [new Application/Traffic/Exponential]
    $exp1($i) set packetSize_ 1500
    $exp1($i) set burst_time_ 100ms
    $exp1($i) set idle_time_ 400ms
    $exp1($i) set rate_ 1000000
    $exp1($i) attach-agent $udp1($i)
    $ns at 0.$i "$exp1($i) start"
}

# class 3
for {set i [expr ($val(numFlowPON1) + 1)]} {$i <= [expr $val(numFlowP3N1) + $val(numFlowPON1)]} {incr i} {
    set sink1($i) [new Agent/TCPSink]
    $ns attach-agent $n(0) $sink1($i)
}
for {set i [expr ($val(numFlowPON1) + 1)]} {$i <= [expr $val(numFlowP3N1) + $val(numFlowPON1)]} {incr i} {
    set tcp1($i) [new Agent/TCP/Newreno]
    $tcp1($i) set packetSize_ 1500
    $tcp1($i) set class_ 3
    $tcp1($i) set prio_ 3
    $ns attach-agent $n(1) $tcp1($i)
    puts "tcp1($i) class 3 node created ..."
}
for {set i [expr ($val(numFlowPON1) + 1)]} {$i <= [expr $val(numFlowP3N1) + $val(numFlowPON1)]} {incr i} {
    $ns connect $tcp1($i) $sink1($i)
}
for {set i [expr ($val(numFlowPON1) + 1)]} {$i <= [expr $val(numFlowP3N1) + $val(numFlowPON1)]} {incr i} {
    set ftp1($i) [new Application/FTP]
    $ftp1($i) set packetSize_ 1500
    $ftp1($i) attach-agent $tcp1($i)
    $ns at 0.$i "$ftp1($i) start"
}

# Traffic sources n0
# class 0
for {set i 1} {$i <= $val(numFlowPON0)} {incr i} {
    set null0($i) [new Agent/Null]
    $ns attach-agent $n(1) $null0($i)
}
for {set i 1} {$i <= $val(numFlowPON0)} {incr i} {
    set udp0($i) [new Agent/UDP]
    $udp0($i) set packetSize_ 1500
    $udp0($i) set class_ 0
    $udp0($i) set prio_ 0
    $ns attach-agent $n(0) $udp0($i)
    puts "udp0($i) class 0 node created ..."
}
for {set i 1} {$i <= $val(numFlowPON0)} {incr i} {
    $ns connect $udp0($i) $null0($i)
}
for {set i 1} {$i <= $val(numFlowPON0)} {incr i} {
    set exp0($i) [new Application/Traffic/Exponential]
```

```

    $exp0($i) set packetSize_ 1500
    $exp0($i) set burst_time_ 100ms
    $exp0($i) set idle_time_ 400ms
    $exp0($i) set rate_ 1000000
    $exp0($i) attach-agent $udp0($i)
    $ns at 0.5$i "$exp0($i) start"
}

# class 3
for {set i [expr ($val(numFlowPON0) + 1)]} {$i <= [expr $val(numFlowP3N0) + $val(numFlowPON0)]} {incr i} {
    set sink0($i) [new Agent/TCPSink]
    $ns attach-agent $n(1) $sink0($i)
}
for {set i [expr ($val(numFlowPON0) + 1)]} {$i <= [expr $val(numFlowP3N0) + $val(numFlowPON0)]} {incr i} {
    set tcp0($i) [new Agent/TCP/Newreno]
    $tcp0($i) set packetSize_ 1500
    $tcp0($i) set class_ 3
    $tcp0($i) set prio_ 3
    $ns attach-agent $n(0) $tcp0($i)
    puts "tcp0($i) class 3 node created ..."
}
for {set i [expr ($val(numFlowPON0) + 1)]} {$i <= [expr $val(numFlowP3N0) + $val(numFlowPON0)]} {incr i} {
    $ns connect $tcp0($i) $sink0($i)
}
for {set i [expr ($val(numFlowPON0) + 1)]} {$i <= [expr $val(numFlowP3N0) + $val(numFlowPON0)]} {incr i} {
    set ftp0($i) [new Application/FTP]
    $ftp0($i) set packetSize_ 1500
    $ftp0($i) attach-agent $tcp0($i)
    $ns at 0.2$i "$ftp0($i) start"
}

#=====
## Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(SimTime) "$n($i) reset";
}
$ns at $val(SimTime) "stop"
$ns at [expr $val(SimTime)] "puts \"NS EXITING...\" ; $ns halt"
proc stop {} {
    global ns tracefd
    $ns flush-trace
    close $tracefd
#    exec nam 3.nam &
}

$ns at 240 "sendpacket"

puts "Starting Simulation..."
$ns run

```

Esta página ha sido intencionalmente dejada en blanco.

# Referencias

- [1] A. Martínez, V. Villarroel, J. Seoane, F. del Pozo, *Analysis of Information and Communication needs in Rural Primary Healthcare in Developing Countries*, IEEE Transactions on Information Technology in Biomedicine, Vol. 9, No. 1, Mar 2005
- [2] Equipos de Videoconferencia para utilizar en Telemedicina y Teleeducación, [en línea]: [http://www.lifesizehd.es/Industry\\_Solutions/Healthcare\\_Solutions/Telemedicine\\_in\\_HD.aspx](http://www.lifesizehd.es/Industry_Solutions/Healthcare_Solutions/Telemedicine_in_HD.aspx)
- [3] Equipos de Videoconferencia y otras aplicaciones para usar en Teleeducación, [en línea]: <http://www.cisco.com/web/strategy/education/index.html>
- [4] Telemedicina en Perú, [en línea]: <http://perusmart.com/2012/08/21/telemedicina-una-realidad-en-peru/>
- [5] Telemedicina en Perú, [en línea]: <http://telemedicinaandreapinzon.blogspot.com/>
- [6] Aravind eye care system, [en línea]: <http://www.aravind.org>
- [7] Plan Ceibal - Uruguay, [en línea]: <http://www.ceibal.org.uy>
- [8] *Ceibal en Inglés*, [en línea]: <http://www.presidencia.gub.uy/wps/wcm/connect/-presidencia/portalpresidencia/comunicacion/comunicacionnoticias/ceibal-en-ingles>
- [9] *IEEE, IEEE Standard 802.11, 2007 Edition*, [en línea]: <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>
- [10] F.J. Simó-Reigadas, A. Martínez-Fernández, F.J. Ramos-López and J. Seoane-Pascual, *Modeling and optimizing IEEE 802.11 DCF for long-distance links*, IEEE Transactions on Mobile Computing. 9(6):881-896, Jun. 2010
- [11] A. Rendón, A. Martínez, M. F. Dulcey, J. Seoane, R. G. Shoemaker, V. Villarroel, D. M. López, J. Simó, *Rural Telemedicine Infrastructure and Services in the Department of Cauca, Colombia*, Telemedicine Journal and e-Health. Volume 11; Number 4; 2005. pp 451 - 459. ISSN: 1530-5627. (IF 0,817 en 2005) (38 de 53 en la categoría Health Care Sciences & Services).

## Referencias

- [12] S. Nedeveschi, R. K. Patra, S. Surana, S. Ratnasamy, L. Subramanian, E. Brewer, *An adaptive, high performance mac for long-distance multihop wireless networks*, Proceedings of the 14th ACM international conference on Mobile computing and networking, September 14-19, 2008, San Francisco, California, USA.
- [13] R. Patra, S. Nedeveschi, S. Surana, A. Sheth, L. Subramanian, E. Brewer, *WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks*, USENIX NSDI, April 2007
- [14] S. Salmerón, J. Ramiro, A. Martínez, J. Simó, *Traffic engineering in rural wireless networks for developing countries using IEEE 802.11 EDCA*, Wireless Rural and Emergency Communications Conference (IEEE WRECOM), 2007, Roma (Italia).
- [15] J. Simó, A. Martínez, C. Figueras, *Distance Limits in IEEE 802.11 for Rural Networks in Developing Countries*, Wireless Rural and Emergency Communications Conference (IEEE WRECOM), 2007, Roma (Italia).
- [16] *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 Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)), vol., no., pp.0-1-189, 2005 doi: 10.1109/IEEESTD.2005.97890
- [17] *IEEE, IEEE Standard 802.11n*, [en línea]: <http://standards.ieee.org/getieee802/download/802.11n-2009.pdf>, 2009.
- [18] Deyun Gao, Jianfei Cai, King Ngi Ngan, *Admission Control in IEEE 802.11e Wireless LANs*, IEEE Network Magazine, July/August 2005
- [19] Y. Liu, M. Meng, *Survey of admission control algorithms in IEEE 802.11e Wireless LANs*, 2009 ETP International Conference on Future Computer and Communication.
- [20] *IEEE 802.16. 802.16-2009 IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile wireless access systems*, 2009.
- [21] G. Bianchi, *Performance Analysis of IEEE 802.11 Distributed Coordination Function*. *IEEE Journal on Selected Areas in Communications*, 18(3):535-547, Mar. 2000.
- [22] P. Serrano, A. Banchs, P. Patras, A. Azcorra, *Optimal Configuration of 802.11e EDCA for Real-Time and Data Traffic*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 59, NO. 5, JUNE 2010

- [23] K. Huang, K.R. Duffy, D. Malone, *On the Validity of IEEE 802.11 MAC Modeling Hypotheses*, Networking, IEEE/ACM Transactions on , vol.18, no.6, pp.1935-1948, Dec. 2010 doi: 10.1109/TNET.2010.2051335
- [24] A. Herms, G. Lukas, *Preventing admission failures of bandwidth reservation in wireless mesh networks*, Computer Systems and Applications, ACS/IEEE International Conference on, 0:10941099, 2008.
- [25] C. Rattaro, P. Belzarena, *Throughput prediction in wireless networks using statistical learning*, Latin-American Workshop on Dynamic Networks. Buenos Aires - 2010
- [26] *Tecnologías de la Información y las Comunicaciones para zonas rurales. Aplicación a la atención de salud en países en desarrollo*, ISBN 978-84-15413-08-0, CYTED, 2012
- [27] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet Architecture: An Overview*, RFC 1633, June 1994.
- [28] S. Blake et al., *An Architecture for Differentiated Service*, RFC 2475, Dec. 1998.
- [29] R. Agila, J. Sánchez, *Análisis de estándares 802.11e y 802.11n para largas distancias*, Tesis de Grado (Universidad Técnica Particular de Loja)
- [30] S. Nor, A. Mohd, C.Cheow, *An Admission Control Method for IEEE 802.11e*, Network Theory and Applications, 2006.
- [31] A. Andreadis, G. Benelli, R. Zambon, *An Admission Control Algorithm for QoS Provisioning in IEEE 802.11e EDCA*, 3rd ISWPC. Santorini, Greece, 2008
- [32] X. Chen, H. Zhai, X. Tian, *Supporting QoS in IEEE 802.11e Wireless LANs*, IEEE Transactions on Wireless LANs Communications, 2009
- [33] *Siyakhula Living Lab*, [en línea]: <http://siyakhulall.org/>
- [34] *Wimax Forum*, [en línea]: <http://www.wimaxforum.org>
- [35] N. Mendiola Almaraz, Tesis de Maestría, *Estrategia de integración para equipos 802.11e-edca (wifi) y 802.16 (wimax) con soporte para qos*, 2010.
- [36] Byeong Gi Lee, Sunghyun Choi, *Broadband Wireless Access and Local Networks Mobile WiMAX and WiFi*
- [37] K. Duffy, D. Malone, D. J. Leith, *Modeling the 802.11 Distributed Coordination Function in non-saturated conditions*, IEEE Comm. Lett., 9(8):715–717, 2005.
- [38] M. Ergen, P. Varaiya, *Throughput analysis and admission control in IEEE 802.11a*, ACM-Kluwer MONET, 10(5):705–716, 2005.

## Referencias

- [39] F. Maguolo, M. Lacage, T. Turetti, *Efficient collision detection for auto rate fallback algorithm*, Computers and Communications, 2008. ISCC 2008. IEEE Symposium on , vol., no., pp.25-30, 6-9 July 2008 doi: 10.1109/ISCC.2008.4625769
- [40] Y. Li1, C. Wang, X. You, H.-H. Chen, W. She, *Delay and throughput performance of IEEE 802.16 WiMax mesh networks* IET Communications, Junio 2011
- [41] M. Jain, C. Dovrolis, *End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput*, IEEE/ACM Transactions on Networking, Aug. 2003.
- [42] M. Jain , C. Dovrolis, *Pathload: A measurement tool for end-to-end available bandwidth*, Proceedings of Passive and Active Measurements (PAM), 14–25, 2002
- [43] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, L. Cotrell, *PathChirp: Efficient available bandwidth estimation for network paths*, Passive and Active Measurement Workshop, 2003.
- [44] J. Strauss, D. Katabi, F. Kaashoek, *A measurement study of available bandwidth estimation tools*, Internet Measurement Workshop, Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement, 39–44, 2003.
- [45] S. Shah, K. Chen, K. Nahrstedt, *Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks*, in First ISMA/CAIDA Workshop on Bandwidth Estimation (BEst), San Diego, CA, USA, Dec. 2003.
- [46] H.K. Lee, V. Hall, K. H. Yum, K. I. Kim, E. J. Kim1, *Bandwidth Estimation in Wireless Lans for Multimedia Streaming Services*, Advances in Multimedia, Volume 2007, Article ID 70429, 7 pages
- [47] M. Li, M. Claypool, R. Kinicki, *WBest: a bandwidth estimation tool for IEEE 802.11 wireless networks*, in Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN '08), pp. 374–381, October 2008.
- [48] V. N. Vapnik, *The nature of statistical learning theory*, New York, NY, USA: Springer-Verlag New York, Inc., 1995. [en línea]: <http://portal.acm.org/citation.cfm?id=211359>
- [49] G. Gómez, Tesis de Maestría en Ingeniería Eléctrica, *Clasificación de tráfico en Internet utilizando métodos estadísticos*, 2012.
- [50] P. Belzarena, L. Aspirot, *End-to-end quality of service seen by applications: a statistical learning approach* Computer Networks, (March 2010)
- [51] P. Bermolen, D. Rossi, *Support vector regression for link load prediction*, Comput.Netw. (2008), doi:10.1016/j.comnet.2008.09.018

- [52] B. Scholkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Cambridge, MA, USA: MIT Press, 2001.
- [53] B. Scholkopf, A. J. Smola, *A tutorial on support vector regression*, *Statistics and Computing* 14: 199–222, 2004
- [54] H. W. Kuhn, A. W. Tucker, *Nonlinear programming*, in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1950. Berkeley and Los Angeles: University of California Press, 1951, pp. 481-492.
- [55] C-C. Chang, C-J. Lin, *LIBSVM: a library for support vector machines*, [en línea]: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, (2001).
- [56] C. W. Hsu, C. C. Chang, C. J. Lin, *A practical guide to support vector classification*, Taipei, Tech. Rep., 2003. [en línea]: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [57] S. McCanne, S. Floyd S, *ns network simulator*, [en línea]: <http://www.isi.edu/nsnam/ns/>.
- [58] *dei80211mr library*, [en línea]: <http://www.dei.unipd.it/wdyn/?IDsezione=5090>
- [59] S. Wietholter, C. Hoene, *Design and Verification of an IEEE 802.11e ED-CA Simulation Model in ns-2.28*, TKN Telecommunication Network Group Technische Universitat Berlin, November 2003. [en línea]: [http://www.tkn.tu-berlin.de/menue/hardwaresoftware\\_components/software/simulation\\_models/an\\_ieee\\_80211e\\_edcf\\_and\\_cfb\\_simulation\\_model\\_for\\_ns-226/](http://www.tkn.tu-berlin.de/menue/hardwaresoftware_components/software/simulation_models/an_ieee_80211e_edcf_and_cfb_simulation_model_for_ns-226/)
- [60] C. Casetti, *802.11e implementation for the ns simulator*, [en línea]: <http://www.telematica.polito.it/casetti/802.11e>.
- [61] N. Qiang, *ns-802.11e edcf for ieee 802.11e wireless lan*, [en línea]: <http://www-sop.inria.fr/planete/qni/Research.html#ns>.
- [62] A. Andreadis, G. Benelli, R. Zambon, *Evaluation of QoS Support for Multimedia Traffics in IEEE 802.11e*, in *Proc. of SoftCOM, Split-Dubrovnik, Croatia*, September 2006.
- [63] *Technical Report TKN patch*, [en línea]: [http://www.tkn.tu-berlin.de/fileadmin/fg112/Hard\\_Software\\_Components/Software/Simulation-%20Models/techreport.pdf](http://www.tkn.tu-berlin.de/fileadmin/fg112/Hard_Software_Components/Software/Simulation-%20Models/techreport.pdf)
- [64] Weikuo Chu, Yu-Chee Tseng, *Performance Analysis of IEEE 802.11 DCF in a Multi-Rate WLAN*, Department of Computer Science National Chiao-Tung University, Hsin-Chu, Taiwan
- [65] *Instalación de ns-2*, [en línea]: <http://817days.blogspot.com/2011/02/ns228-in-ubuntu-104.html>

## Referencias

- [66] *Albentia Systems S.A.*, ARBA550 BS Series User Guide, [en línea]: <http://www.albentia.com>
- [67] ALIX2D2, [en línea]: <http://www.pcengines.ch/alix2d2.htm>
- [68] ALIX2C0, [en línea]: <http://www.pcengines.ch/alix2c0.htm>
- [69] VOYAGE, [en línea]: <http://linux.voyage.hk/content/voyage-linux-065-released>
- [70] D-ITG, [en línea]: <http://www.grid.unina.it/software/ITG/>
- [71] I. Foche, *Guía introductoria a D-ITG*, [en línea]: [http://wiki.ehas.org/index.php?title=Mediciones\\_con\\_Distributed\\_Internet\\_Traffic\\_Generator](http://wiki.ehas.org/index.php?title=Mediciones_con_Distributed_Internet_Traffic_Generator)
- [72] *Iperf*, [en línea]: <http://dast.nlanr.net/Projects/Iperf/>
- [73] *tcpdump, a powerful command-line packet analyzer; and libpcap, a portable c/c++ library for network traffic capture*, [en línea]: <http://www.tcpdump.org/>
- [74] ITU. ITU-T recommendation Y.1540 (11/07): *IP packet transfer and availability performance parameters*, Technical report, International Telecommunication Union.
- [75] Octave, [en línea]: <http://www.gnu.org/software/octave/>

# Índice de tablas

2.1. Algunas extensiones definidas en IEEE 802.11. . . . .	11
6.1. Resultados de SVM utilizando la media de $K_n$ . . . . .	51
6.2. Resultados SVM. . . . .	54
6.3. Parámetros usados para las dos clases de servicio (EDCA). . . . .	56
6.4. Resumen de resultados al aplicar SVM considerando como estimador del estado de la red la media de $K_n$ . . . . .	62
7.1. Descripción Equipos WiMAX. . . . .	68
7.2. Descripción Equipos WiFi. . . . .	69
7.3. Mapeo entre Categorías de Acceso (AC) y campo DSCP (Driver MadWifi) . . . . .	72
7.4. Parámetros EDCA usados. . . . .	73
7.5. Mapeo entre Categorías de Acceso de EDCA, Flujos de Servicio de WiMAX y campo DSCP. . . . .	74
7.6. Validación de la configuración del segmento WiMAX. . . . .	75
7.7. Resumen de resultados obtenidos al aplicar SVM-C usando la media de $K(n)$ como estimador del estado de la red. . . . .	78

Esta página ha sido intencionalmente dejada en blanco.

# Índice de figuras

1.1. Topología de una escuela con conectividad punto a punto que comparte el acceso a Internet con otra. . . . .	4
2.1. Esquema del funcionamiento de DCF (dos estaciones). Fuente [26]	12
2.2. Comparación 802.11 (DCF) y 802.11e (EDCA). Fuente [29] . . . .	15
3.1. SVR: Margen de ajuste (Fuente: [53]) . . . . .	25
3.2. SVM-C: Separables linealmente. La línea punteada representa el hiperplano buscado. Las líneas enteras representan los márgenes determinados por los SV. . . . .	27
3.3. SVM-C: Caso no linealmente separable. En este caso se mapean las muestras a un espacio de dimensión mayor donde sí son separables mediante un hiperplano. . . . .	27
4.1. Tiempos entre arribos de los paquetes de medida para dos estados diferentes de la red. A: Muchas colisiones y/o alta interferencia. B: Bajas colisiones y baja interferencia. . . . .	32
4.2. Escenario usado para explicar la metodología. . . . .	34
5.1. WLAN, escenario usado para mostrar el comportamiento de $K_n$ al variar distintas condiciones del estado de la red. . . . .	39
5.2. Distribución empírica de $K_n$ para diferentes estados de la red (varía $m$ ) . . . . .	40
5.3. Distribución empírica de $K_n$ para diferentes estados de la red (varía $d$ ). . . . .	40
5.4. Distribución empírica de $K_n$ para diferentes estados de la red (QoS). Los valores de $a$ y $b$ representan la cantidad de conexiones de clase 1 (alta prioridad) y 2 (baja prioridad) respectivamente. . . . .	41
6.1. Topología utilizada para testear la metodología para el caso de predicción de throughput (en condiciones de tráfico saturadas y no saturadas). . . . .	48
6.2. Distribuciones empíricas de $K_n$ para distintos escenarios. Variando la cantidad de clientes $m$ o variando la distancia $d$ entre el NodoX y RN. . . . .	50

## Índice de figuras

6.3. Conjunto 1 de verificación: Predicciones usando sólo la media de $K_n$ como estimador del estado de la red (condiciones de tráfico saturadas). $R = 30$ metros. . . . .	51
6.4. Conjunto 2 de verificación: Predicciones usando sólo la media de $K_n$ como estimador del estado de la red (condiciones de tráfico saturadas). $R = 10$ metros. . . . .	52
6.5. Conjunto 3 de verificación: Predicciones usando sólo la media de $K_n$ como estimador del estado de la red (condiciones de tráfico saturadas). $R = 60$ metros. . . . .	52
6.6. Distribución empírica de $K_n$ para distintos estados del enlace inalámbrico. . . . .	54
6.7. Throughput Simulado y Estimado para el caso de condiciones no saturadas de tráfico usando media, varianza y delay variable mínimo como $X$ . . . . .	55
6.8. Topología I: WLAN 802.11e . . . . .	57
6.9. Topología II: enlace punto a punto . . . . .	57
6.10. Distribución empírica de $K_n$ para diferentes estados de la red. Los valores de $a$ y $b$ representan la cantidad de conexiones de clase 1 y 2 respectivamente (WLAN 802.11e). . . . .	59
6.11. Delay obtenido para la nueva conexión de clase 1 vs Media de $K_n$ . WLAN 802.11e, escenario A. . . . .	60
6.12. Throughput obtenido para la nueva conexión de clase 2 vs Media de $K_n$ . WLAN 802.11e, escenario B. . . . .	61
6.13. Distribución empírica de $K_n$ para diferentes estados de la red. Las curvas se diferencian en la cantidad de flujos de clase 2 que se cursan desde el $AP_1$ al $AP_2$ . Todas las simulaciones se realizan con 5 flujos clase 1 desde el $AP_1$ hacia el $AP_2$ , 5 de clase 1 en sentido contrario y 5 de clase 2 desde el $AP_2$ al $AP_1$ . . . . .	62
6.14. Delay obtenido para la nueva conexión de clase 1 vs Media de $K_n$ . Enlace punto a punto 802.11e, escenario C . . . . .	63
6.15. Throughput obtenido para la nueva conexión de clase 2 vs Media de $K_n$ . Enlace punto a punto 802.11e, escenario D. . . . .	63
6.16. Topología de red donde lo único que varía entre simulaciones es la posición del nodo C1. . . . .	64
6.17. Throughput de la conexión más comprometida (C1-AP) en función de la media de $K_n$ al variar la distancia entre el AP y C1. El NodoX (que es donde se origina la nueva conexión) se mantiene en la misma posición en todas las simulaciones. . . . .	65
7.1. Topología A: Red Híbrida WiMAX ( $f_1=5$ GHz) y WiFi ( $f_2=2,4$ GHz). . . . .	67
7.2. Topología B: Red de Área Local Inalámbrica ( $f=2,4$ GHz). . . . .	68
7.3. Equipo WiMAX usado. . . . .	69
7.4. Hardware usado para los Nodos WiFi . . . . .	70
7.5. Verificación del funcionamiento de EDCA. . . . .	73
7.6. Configuración de los Flujos de Servicio - WiMAX . . . . .	74

7.7. Throughput obtenido por la nueva conexión vs Media de  $K_n$  (Escenario 1) . . . . . 79



Esta es la última página.  
Compilado el 19 de noviembre de 2012.  
<http://iie.fing.edu.uy/>