Clasificación de tráfico en Internet utilizando métodos estadísticos

Gabriel Gómez Sena

Presentado como parte de los requisitos para
obtener el título de Magíster en Ingeniería Eléctrica

Director Académico:

Dr. Pablo Belzarena, Facultad de Ingeniería, Universidad de la República

Directores de Tesis:

Dr. Pablo Belzarena, Facultad de Ingeniería, Universidad de la República

Dra. Paola Bermolen, Facultad de Ingeniería, Universidad de la República

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay
16 de diciembre de 2011

Esta tesis fue preparada en LaTeX
Las gráficas fueron creadas con Gnuplot
Tiene un total de 114 páginas, 67 figuras y 19 tablas.

Versión del 16 de diciembre de 2011

Agradecimientos

A Pablo Belzarena por su constante apoyo y orientación para desarrollar este trabajo.

A Paola Bermolen por su orientación y revisión crítica de este documento.

Al Plan Ceibal por financiar la Beca de Maestría que me otorgó la Comisión Académica de Posgrados de la Facultad de Ingeniería y que me apoyó en la culminación de este trabajo.

A Gabriel Usera y Santiago Iturriaga por el apoyo, asesoramiento y asistencia en la utilización del Cluster de Facultad.

A la Administración Nacional de Telecomunicaciones, ANTEL y al grupo de "Telecommunication Networks" en la Università degli Studi di Brescia, Italia, por brindar acceso a las trazas de tráfico utilizadas en este trabajo.

A todos los integrantes del IIE, de este y de otros tiempos, que me han brindado apoyo y aliento para realizar este trabajo.

A Martín, Guillermo, Juan Manuel y Federico.

A Esther.

Tabla de contenidos

Resumen

Este trabajo propone la identificación de tráfico de Internet en base al análisis estadístico del tamaño de los primeros segmentos intercambiados en las conexiones de transporte.

La identificación de los flujos, analizando solamente el tamaño de los primeros segmentos intercambiados, tiene como principal ventaja su eventual aplicación a la clasificación "en línea" del tráfico. Esta identificación en conjunto con técnicas de calidad de servicio, enrutamiento por políticas, aplanamiento de tráfico, permitiría aportar herramientas para la gestión de una red compleja.

El análisis estadístico propuesto en este trabajo se basa en la utilización de un sistema de aprendizaje supervisado conocido como *Support Vector Machines* (SVM). Esta técnica en los últimos años ha comenzado a aplicarse a la clasificación de tráfico y se caracteriza por su destacada capacidad de discriminación.

De acuerdo a los resultados obtenidos en este trabajo, la precisión de la identificación de tráfico utilizando la técnica SVM caracterizando los flujos por el tamaño de los primeros segmentos intercambiados, es muy buena.

Al analizar el comportamiento de la precisión por tipo de aplicación aparecen diferencias importantes que se mitigaron utilizando la técnica de *boosting*. Esta técnica permite la generación de una secuencia de modelos SVM y se propuso la realización de una votación ponderada entre los resultados de clasificación obtenidos por cada modelo de esta secuencia. Con este procedimiento se logró incrementar la precisión de la clasificación para aquellas aplicaciones que obtenían menores niveles con la técnica de SVM, sin perjuicio de las restantes, lográndose una mejora en los resultados obtenidos en primera instancia.

1 Introducción

1.1. Motivación

Internet es una red de comunicación de datos que brinda soporte para el intercambio de información entre los equipos que la integran.

Por su concepción de diseño, la capa de red en Internet, permite la comunicación entre dos nodos extremos (a nivel de plano de datos) mediante un mecanismo relativamente simple que requiere baja capacidad de procesamiento en los nodos intermedios (enrutadores o *routers*). La complejidad de la red reside en los extremos (*end nodes*), los cuales implementarán las capas superiores (transporte y aplicación) necesarias para lograr una comunicación bajo control del usuario.

Estas consideraciones de diseño han posibilitado la prevalencia de Internet como red global de transporte de datos ya que su arquitectura permite y promueve que los usuarios desarrollen sus propias aplicaciones. Eventualmente cada aplicación diseñará su propio protocolo de comunicación a nivel de capa de aplicación. El éxito de cada aplicación dependerá de la difusión y penetración que logre en el conjunto de usuarios de Internet. Algunas aplicaciones han logrado despertar masivamente el interés de los usuarios y otras no lo han logrado, o lo han logrado por períodos de tiempo acotados.

El tipo de servicio que brinda la capa de red en Internet se basa en un criterio de mejor esfuerzo (*best effort*) que puede resumirse en que la red "hará su mejor esfuerzo" para que las unidades de datos (paquetes) lleguen a destino. Esto implica que la red controla en forma muy débil el tráfico que generan sus usuarios lo que naturalmente incentiva a que estos últimos intenten obtener el mayor beneficio de la misma. A modo de ejemplo, es común que los usuarios que contratan un acceso a Internet con tarifa plana, busquen la forma de utilizar ese enlace todo el tiempo a su máxima capacidad, utilizando incluso aplicaciones optimizadas para mejorar la performance de las descargas de archivos o videos.

Se genera entonces una tensión entre los recursos que el usuario desea

obtener de la red y los que la red está en condiciones de brindarle, lo cual ha llevado al desarrollo de métodos para controlar el tráfico generado por los usuarios. Esta ingeniería de tráfico, a modo de ejemplo, puede implementarse mediante políticas de encolamiento y descarte de paquetes en los enrutadores así como a través de políticas de aplanamiento de tráfico (*traffic shaping*). Estas políticas son necesarias para un correcto funcionamiento de la red global ya que permiten controlar la congestión que podría producirse al excederse la capacidad de tráfico que la red puede procesar con normalidad.

En muchos casos puede ser de interés la aplicación de políticas de ingeniería de tráfico específicas para determinados tipos de tráfico que generen una demanda excesiva sobre la red, o que simplemente tengan requerimientos específicos. Por ejemplo, si se desea mejorar la calidad de las llamadas de voz sobre IP[1], podría encaminarse el tráfico de este servicio a través de enlaces de menor retardo, lo que requeriría la aplicación de políticas de enrutamiento específicas. Otras aplicaciones podrían requerir diferentes calidades de servicio (QoS[2]) que podrían implementarse mediante mecanismos existentes en Internet (Diffserv[3], MPLS[4], etc). Estos mecanismos requieren una asociación del tráfico a diferentes clases –correspondientes a diferentes calidades de servicio– que luego serán tratadas con políticas de encaminamiento o encolamiento diferenciadas.

Aún en el caso de no requerirse la aplicación de políticas específicas a los diferentes tipos de tráfico, es importante para el operador de la red, disponer de un conocimiento detallado de la composición del tráfico que circula por su red como insumo para la planificación del crecimiento y eventualmente para el ajuste del diseño de la misma.

La identificación de los tipos de tráfico en la red también puede ser de utilidad para la detección de anomalías en el contexto del análisis de seguridad de la red. Mediante sistemas de monitorización que detecten los cambios en la composición del tráfico de la red podrían identificarse comportamientos anómalos en dicha composición, por ejemplo debidos a un crecimiento no esperado de algún tipo de tráfico.

En resumen, la identificación y clasificación del tráfico de Internet es de interés en múltiples actividades relacionadas con la planificación y operación de las redes complejas. Por estos motivos, este trabajo aborda la identificación y clasificación de tráfico en Internet, apuntando a la obtención de mecanismos que permitan aplicar políticas de enrutamiento, aplanamiento,

[1] Internet Protocol
[2] Quality of Service
[3] Differentiated Services
[4] Multiprotocol Label Switching

asignación de clases de servicio y que redunden en un mejor funcionamiento de la red global. Para cumplir estos objetivos, serán necesarios mecanismos que permitan identificar el tráfico tan pronto como sea posible a efectos de poder tomar alguna de las acciones mencionadas. Será deseable entonces que, los mecanismos que se propongan, puedan identificar el tráfico en base a poca información y de forma eficiente (con un costo computacional reducido).

1.2. Síntesis de la propuesta

En este trabajo se propone la identificación de tráfico de Internet en base al análisis estadístico del tamaño de los segmentos de las conexiones de transporte.

El conocido método de clasificación en base a la búsqueda de patrones en el contenido de los datos de capa de aplicación (DPI[5]) reseñado en el capítulo 2, analiza solamente el contenido de los primeros bytes del flujo de datos de usuario a efectos de identificar el tráfico. La fundamentación de esta decisión se basa en que las aplicaciones realizan al comienzo del flujo ciertos intercambios característicos que permiten identificarlas y por lo tanto alcanza con buscar los patrones específicos en los primeros bytes intercambiados. Esta misma justificación permite pensar que la identificación de tráfico mediante el análisis estadístico del tamaño de los segmentos podría realizarse considerando solamente los primeros segmentos intercambiados. Por otra parte, la información requerida para este análisis no presenta las debilidades de la técnica DPI en cuanto al acceso a información eventualmente sensible de los usuarios.

Si es posible identificar la aplicación transportada por un flujo analizando solamente el tamaño de los primeros segmentos intercambiados, se podría utilizar esa identificación para realizar una clasificación "en línea" del tráfico. A modo de ejemplo, una vez identificado el tráfico, se podría comenzar a marcar los paquetes de ese flujo con una etiqueta que luego fuera tenida en cuenta por algún mecanismo de calidad de servicio, asignando diferentes colas de procesamiento en los enrutadores a diferentes clases de tráfico (a diferentes etiquetas). También sería posible aplicar políticas de enrutamiento diferenciadas para diferentes clases de tráfico.

El análisis estadístico propuesto en este trabajo se basa en la utilización de un sistema de aprendizaje supervisado basado en la técnica de *Support Vector Machines* (SVM) que se presenta en el capítulo 4. Esta técnica es usada en otros campos de aplicación y en los últimos años ha comenzado

[5]Deep Packet Inspection

a proponerse para la clasificación de tráfico. Resulta de interés para este trabajo ya que se caracteriza por su destacada capacidad de discriminación, así como su bajo costo computacional.

Las técnicas de aprendizaje estadístico supervisado requieren una fase de entrenamiento y una fase de prueba o verificación. A partir de un conjunto de muestras pre-clasificadas se destina una parte de ese conjunto para entrenar un modelo y el resto para verificarlo. Analizando el resultado de la clasificación obtenida para las muestras de verificación, puede calcularse la precisión obtenida por el método utilizado.

De acuerdo a los resultados obtenidos en este trabajo y presentados en el capítulo 6, la precisión de la identificación de tráfico utilizando la técnica SVM cuando se caracterizan los flujos por el tamaño de los primeros segmentos intercambiados, es alta. Sin embargo, analizando el comportamiento con mayor detalle, se observa que la precisión en la clasificación puede ser muy diferente para las distintas aplicaciones.

Por lo expresado anteriormente, este trabajo propone un mecanismo para mejorar la precisión en la clasificación con SVM para aquellas aplicaciones para las que se obtuvieron índices de precisión menores. El mecanismo propuesto es la utilización de la técnica de *boosting* (ver sección 4.1.2) para la generación de una secuencia de modelos SVM, realizando posteriormente una votación ponderada entre los veredictos de los modelos generados. Esta técnica permitió incrementar la precisión de las aplicaciones mencionadas, sin perjuicio de las restantes. El mecanismo de votación propuesto se detalla en la sección 5.2.3.

1.3. Principales resultados de este trabajo

En este trabajo se alcanzaron los siguientes resultados:

- Se realizó una revisión bibliográfica del estado actual de la investigación en el área de clasificación e identificación de aplicaciones.
- Se verificó la posibilidad de identificación de tráfico tomando en cuenta solamente el tamaño de los primeros segmentos de un flujo de capa de transporte. Se analizó el comportamiento tanto para TCP[6] como para UDP[7])

[6]Transport Control Protocol
[7]User Datagram Protocol

- Se verificó que con la descripción de flujos utilizada, la técnica de *Support Vector Machines* brinda altos niveles de precisión. Se realizó un exhaustivo análisis de la variación de la precisión de los resultados en función de los parámetros involucrados (cantidad de segmentos, tamaño de la muestra de entrenamiento, parámetros del modelo SVM)
- Se logró mejorar la precisión obtenida con la técnica de SVM mediante un mecanismo de *boosting* para la generación de una secuencia de modelos y aplicando un sistema de votación ponderada entre los veredictos de cada modelo.
- Algunos de los resultados de este trabajo fueron publicados en [1].

1.4. Organización de este documento

Para su mejor comprensión, este documento se ha organizado de la siguiente forma:

En el capítulo 2 se analizan las diferentes técnicas de identificación y clasificación de tráfico propuestas en la bibliografía relevada. Se detallan sus métodos, fortalezas y debilidades. Se mencionan las posibles aplicaciones de los sistemas de identificación y clasificación de tráfico a la gestión de calidad de servicio y a la ingeniería de tráfico.

En el capítulo 3 se exponen las principales decisiones surgidas a partir del análisis de la bibliografía relevada.

Posteriormente en el capítulo 4 se realiza una reseña de las herramientas estadísticas utilizadas, brindando una breve introducción a las mismas.

En el capítulo 5 se presenta la metodología utilizada, brindando detalles sobre la implementación de los métodos de clasificación propuestos, así como los datos y medidas utilizados para validarlos.

Finalmente, en el capítulo 6 se analizan los resultados obtenidos con los diferentes métodos analizados, presentándose en el capítulo 7 los principales resultados y algunas posibles líneas de trabajo futuro.

2 Clasificación de tráfico

En este capítulo se analizan las diferentes metodologías de clasificación de tráfico en Internet que aparecen en la revisión bibliográfica realizada, identificando sus principales ventajas y debilidades. Posteriormente se analizan diferentes escenarios en los que los mecanismos de identificación y clasificación pueden ser de utilidad.

A partir del análisis crítico de las propuestas existentes, en el capítulo 3 se justifican los criterios adoptados para el desarrollo de este trabajo.

2.1. Flujos en capa de transporte

En la arquitectura de Internet, las aplicaciones de usuario utilizan principalmente los protocolos de transporte TCP y UDP para intercambiar sus datos. A nivel de capa de transporte se identifican por tanto flujos de datos entre entidades pares, residentes en los equipos de usuario que intercambian información. Estos flujos de datos se identifican por 5 parámetros presentes en los encabezados:

- Protocolo de capa de transporte (campo *Protocol* del encabezado IP)
- Dirección IP de origen (campo *Source Address* del encabezado IP)
- Dirección IP de destino (campo *Destination Address* del encabezado IP)
- Puerto de origen (campo *Source Port* del encabezado TCP o UDP)
- Puerto de destino (campo *Destination Port* del encabezado TCP o UDP)

Para el problema que motiva este trabajo, interesa determinar qué aplicación de usuario es la que está generando los datos transportados por cada flujo.

2.2. Métodos de clasificación de tráfico

2.2.1. Análisis de los encabezados de capa de transporte

El método más antiguo para la clasificación de tráfico se basa en el análisis de la información contenida en los encabezados del protocolo de transporte. Esto es, analizar el campo de puerto destino (*Destination Port*) del protocolo TCP o UDP de alguno de los segmentos de un flujo. El método es muy simple ya que solamente requiere analizar el valor de un campo y en base a ese valor (número de 16 bits) decidir qué protocolo transporta ese flujo.

Este método ha sido utilizado históricamente por su simplicidad, y su precisión se basa en el supuesto que cada aplicación tiene asociado un número de puerto de transporte. Si bien existe una lista de puertos bien conocidos (*well known ports*) asignados por la IANA[1], muchas aplicaciones actuales no registran sus puertos ante la IANA.

Por otra parte, algunas aplicaciones utilizan negociación dinámica de puertos o utilizan puertos asignados a otras aplicaciones. Es típico que muchas aplicaciones actuales utilicen el puerto 80 (asignado por IANA al protocolo HTTP[2]) para transportar otros protocolos. Este comportamiento se explica por las restricciones normalmente existentes en las redes empresariales y de otras instituciones, donde es frecuente el uso de corta fuegos (*firewalls*) como elementos de seguridad. Utilizando el puerto 80, estas aplicaciones aparecen a la red como clientes de navegación estándar y de este modo pueden pasar inadvertidas por los elementos de seguridad de la red.

Es por estos motivos que, siendo el análisis de los puertos un método simple y eficiente, ha dejado de utilizarse debido a su baja precisión en la clasificación [2, 3, 4].

2.2.2. Análisis del contenido de capa de transporte

Este método consiste en analizar el contenido de la carga útil (*payload*) de los protocolos de transporte mediante una búsqueda de patrones. El método se sustenta en que las diferentes aplicaciones al comienzo del flujo realizan ciertos intercambios mediante instrucciones específicas, ya sea para negociar opciones o para especificar acciones a realizar (comandos de control). Por tanto, solamente es necesario buscar patrones conocidos dentro de los pri-

[1] Internet Assigned Numbers Authority, http://www.iana.org/
[2] Hypertext Transfer Protocol

meros bytes de la carga útil de un flujo para determinar qué protocolo de capa de aplicación transporta. A modo de ejemplo, el siguiente intercambio puede apreciarse en el principio de un flujo TCP generado por un navegador al solicitar una página web a un servidor utilizando el protocolo HTTP. Se muestra el principio del intercambio cuando el navegador solicita la página y antes que la misma sea entregada por el servidor.

GET http://www.example.com HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 10 May 2011 22:30:28 GMT
Server: Apache
Last-Modified: Tue, 10 May 2011 22:04:06 GMT
ETag: "8c075dc96350ead8476dbd48d0deb29f"
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Cache-Control: must-revalidate
Content-Type: text/html; charset=utf-8
Set-Cookie: SESSb0b18b3b4480e1cffa8e25a4a48c4fa3=hp1hito7gikp273vs4q5tm9t77;
expires=Fri, 03-Jun-2011 02:09:05 GMT; path=/; domain=example.com
Connection: close

Se puede apreciar en este extracto, la existencia de ciertos patrones característicos que podrán utilizarse para indentificar la aplicación generadora de este tráfico.

Este método, conocido como *DPI*, obtiene buena precisión en la clasificación y permite identificar aplicaciones en base a su comportamiento e independientemente del puerto de transporte que utilicen. Sin embargo es conocido que la búsqueda de patrones es una tarea computacionalmente costosa y pueden requerirse expresiones regulares bastante complejas para identificar algunos protocolos de capa de aplicación.

Por otra parte este método ha sido cuestionado ya que requiere acceso a la carga útil de los segmentos de transporte donde eventualmente podría viajar información sensible para el usuario; por ejemplo nombres de usuario, contraseñas, números de tarjetas de crédito, páginas visitadas, etc. Otra de las limitaciones de este método es que no es aplicable a tráfico encriptado ya que la información necesaria para la búsqueda de patrones no es accesible [2, 5, 6, 7, 8, 9, 10].

Por estos motivos se han continuado desarrollando métodos alternativos para la identificación de aplicaciones.

2.2.3. Análisis del comportamiento de los equipos

Recientemente se han desarrollado técnicas de clasificación o identificación de tráfico basadas en un análisis del comportamiento de los equipos. Estos métodos apuntan, por ejemplo, a diferenciar en forma gruesa tráfico *peer-to-peer (p2p)*[3] de no *p2p*, teniendo en cuenta que típicamente un equipo que está ejecutando una aplicación *p2p* abre más de una conexión simultánea hacia varios servidores, muchas veces estableciendo flujos TCP y UDP simultáneamente. Si un equipo se comporta de esta manera, puede asumirse que está intercambiando tráfico del tipo *peer-to-peer*. Por otra parte si se encuentra un equipo que recibe muchas conexiones desde diferentes orígenes, pero todas ellas destinadas al puerto 25 (asignado al protocolo SMTP[4]), podría concluirse que se está en presencia de un servidor de correo [4, 7].

Estos mecanismos requieren un análisis complejo por equipo debiéndose mantener un estado de las conexiones originadas o recibidas por cada equipo y no son aplicables al problema objeto de este trabajo.

2.2.4. Análisis estadístico de propiedades de los flujos

En los últimos años aparecen en la literatura varias propuestas de clasificación de tráfico basadas en el análisis estadístico de algunas propiedades de los flujos.

Características de los flujos tales como el tiempo entre arribo de los segmentos, la cantidad de bytes totales transportados por el flujo, la cantidad de segmentos intercambiados en uno o ambos sentidos de la comunicación, la duración total del flujo, la duración de los intervalos de tiempo de silencio, el tamaño de los segmentos intercambiados y otras propiedades, han sido analizadas en cuanto a su aplicabilidad a la identificación de tráfico en varios trabajos [2, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21].

Para el análisis estadístico se utilizan técnicas de aprendizaje estadístico (*machine learning*) que pueden ser supervisadas, no supervisadas o semi supervisadas. Las técnicas no supervisadas o semi supervisadas, proponen mecanismos de generación automática o semi automática de categorías o clases de tráfico [22, 23]. Estas técnicas permiten identificar tipos de tráfico que se comportan estadísticamente de modo similar. Sin embargo, las propuestas más frecuentes en la literatura proponen la utilización de mecanismos

[3] Aplicaciones entre pares en contraposición a la clásica arquitectura cliente-servidor. Típicamente utilizadas para descargar archivos, música, etc

[4] Simple Mail Transfer Protocol

del tipo supervisado. Las técnicas de aprendizaje estadístico supervisado se basan en realizar una primer fase de entrenamiento o aprendizaje, en la cual se ajusta un modelo y posteriormente una fase de prueba o verificación, en la cual se analiza la calidad del modelo generado [18, 24, 25].

Este tipo de análisis, basado en las características mencionadas anteriormente, presenta algunas fortalezas interesantes. Al basarse en el estudio de propiedades de los flujos, no es necesario acceder a la carga útil transportada por el flujo y por tanto no se violenta la privacidad de los usuarios. Adicionalmente y por este mismo motivo, el análisis estadístico es aplicable a tráfico encriptado y permitirá identificar además tráfico que utilice puertos dinámicos o asignados a otras aplicaciones ya que la información de puertos no tiene por que incluirse en el análisis.

2.3. Escenarios de aplicación de la clasificación de tráfico

Como ya se mencionó, existen diferentes escenarios donde es de interés contar con técnicas de identificación y clasificación de tráfico.

Si se requiere la aplicación de mecanismos para brindar calidad de servicio (QoS) a diferentes aplicaciones, será necesario asignar al tráfico generado por cada aplicación un identificador o etiqueta que lo asocie a una determinada clase de servicio. En base a esa identificación el tráfico será posteriormente tratado por algún mecanismo que provea la calidad de servicio requerida. A modo de ejemplo, diferentes clases de tráfico podrán asociarse a diferentes colas de procesamiento en los enrutadores, podrán encaminarse por diferentes caminos MPLS o por diferentes rutas a nivel IP, brindando así un tratamiento diferenciado a cada clase de tráfico. Es claro que para la gestión de calidad de servicio, se requiere detectar el tipo de tráfico lo antes posible, idealmente apenas comienza el flujo, de modo de poder tomar acciones antes que el flujo haya finalizado. Este escenario es también el requerido cuando interesa aplicar mecanismos de aplanamiento de tráfico (*traffic shaping*) o cuando la identificación de aplicaciones se utiliza para la detección de anomalías en el tráfico de la red, donde se requiere tomar alguna acción de forma inmediata. En estos escenarios se requiere naturalmente un tipo de clasificación en línea (*on-line*) y por tanto la identificación de los flujos deberá realizarse tempranamente [26, 27, 28]. Esto significa que una vez detectado el comienzo de un flujo, deberá utilizarse un mecanismo de respuesta rápida y de bajo costo computacional para identificar el flujo y poder tomar acciones sobre el mismo.

En otros escenarios como el análisis forense del tráfico o el análisis de

la evolución histórica del tráfico, no se requiere la identificación en línea y por tanto pueden analizarse otras propiedades de los flujos –por ejemplo la duración– que no son aplicables a la clasificación en línea [13, 29].

Es importante notar que cualquier mecanismo de clasificación de tráfico que pretenda modificar el comportamiento del tráfico en el sentido de filtrar, aplanar o realizar algún tratamiento diferenciado del mismo, puede ser objeto de objeciones en cuanto a la violación del principio de "neutralidad de la red" ("*net neutrality*") [30]. Este principio se basa en que se debe garantizar el acceso universal e indiscriminado a la red en el sentido de que el usuario debe tener la posibilidad de acceder a cualquier servicio o contenido legal en Internet y que recíprocamente tenga el derecho a que sus recursos estén accesibles para cualquier otro usuario de Internet. Sin embargo los proveedores de servicio utilizan mecanismos de gestión del ancho de banda que normalmente pueden incluir políticas de priorización y aplanamiento de tráfico, generalmente sin llegar al bloqueo del mismo.

Por tanto, el problema de la clasificación de tráfico es de interés para los proveedores de servicio y los mecanismos que se propongan podrían utilizarse solamente a efectos de conocer el tráfico circulante por la red y así poder prever las inversiones necesarias, sin necesariamente alterar el comportamiento de la red.

3 Detalle de la propuesta

Los métodos de clasificación que buscan patrones en el contenido de los datos de capa de aplicación (DPI), referidos en el capítulo 2, típicamente realizan la búsqueda solamente en los primeros bytes de un flujo de datos de usuario. Esta decisión se justifica en el hecho que al comienzo del flujo es cuando las aplicaciones realizan ciertos intercambios característicos. Esta misma justificación permite suponer que la identificación de tráfico mediante el análisis estadístico del tamaño de los segmentos podría realizarse analizando solamente los primeros segmentos intercambiados. Si se logra este objetivo, se puede utilizar esa identificación para realizar una clasificación "en línea" del tráfico.

Por lo antedicho, en este trabajo se decidió analizar la precisión de la clasificación de tráfico en base a la información contenida en el tamaño de los primeros segmentos intercambiados por un flujo de capa de transporte, de modo que las soluciones propuestas puedan eventualmente aplicarse para la identificación de tráfico en línea.

Como herramienta de análisis estadístico se propone la utilización de un sistema de aprendizaje estadístico supervisado mediante la técnica de *Support Vector Machines* (SVM) que se presenta en detalle en el capítulo 4. Esta técnica, aplicada desde hace años para resolver otros problemas de clasificación y regresión, aparece en en los últimos años propuesta como herramienta aplicable también a la clasificación de tráfico. Esta técnica se caracteriza por su destacada capacidad de discriminación y su alto desempeño.

Como se detalla en el capítulo 6, la precisión global de la identificación de tráfico utilizando la técnica SVM aplicada al tamaño de los primeros segmentos intercambiados por un flujo, es como mínimo del 80 %. Sin embargo, cuando se analiza su comportamiento con mayor detalle, se observa que la precisión en la clasificación puede ser muy variable para las distintas aplicaciones. Por este motivo, se propone un mecanismo para mejorar la precisión en la clasificación de aquellas aplicaciones para las que se obtuvieron índices de precisión menores. El mecanismo propuesto incluye la utilización de la técnica de *boosting*, descrita en la sección 4.1.2, que brinda una metodo-

logía para la generación de una secuencia de modelos SVM. La clasificación se realizará aplicando una votación ponderada entre los veredictos de cada uno de los modelos SVM generados. Esta técnica permitió incrementar la precisión de las aplicaciones que obtuvieron menores niveles de precisión en la clasificación con un único modelo SVM, sin perjuicio de las restantes.

En lo que resta de este capítulo se definen los indicadores utilizados en el trabajo y se detallan las referencias bibliográficas que fundamentan las decisiones adoptadas.

3.1. Definición de indicadores

A efectos de analizar el comportamiento de los clasificadores y poder comparar las diferentes propuestas se utilizan en la literatura algunos indicadores típicos [31]:

- **Falsos negativos** (FN): porcentaje de muestras que pertenecen a la clase X y se clasifican incorrectamente como no pertenecientes a la clase X.
- **Falsos positivos** (FP): porcentaje de muestras que no pertenecen a la clase X y se clasifican incorrectamente como pertenecientes a la clase X.
- **Verdaderos negativos** (TN): porcentaje de muestras que no pertenecen a la clase X y se clasifican correctamente como no pertenecientes a la clase X.
- **Verdaderos positivos** (TP): porcentaje de muestras que pertenecen a la clase X y se clasifican correctamente como pertenecientes a la clase X. Este indicador se denomina también **confiabilidad**.
- **Precisión**: porcentaje de muestras bien clasificadas sobre las muestras totales.

A partir de las definiciones indicadas, se puede observar que:

$$TP = 100\,\% - FN$$

$$TN = 100\,\% - FP$$

$$\textbf{Precisión} = \frac{TP + TN}{TP + TN + FP + FN}$$

La **precisión** es el indicador mayoritariamente utilizado en los trabajos referenciados y es por tanto el que se adoptó en este trabajo para comparar los diferentes métodos analizados.

3.2. Tamaño de primeros segmentos intercambiados

Varios de los trabajos analizados en los que se proponen mecanismos de identificación o clasificación de tráfico mediante el análisis de estadístico de algunas propiedades de los flujos, coinciden en que una de las propiedades con mayor incidencia en la precisión de la clasificación es el tamaño de los segmentos intercambiados en un flujo [1, 26, 27, 32, 33, 34, 35, 36]. Particularmente para la clasificación de tráfico en línea, interesa identificar el tráfico tan pronto como sea posible, de modo de poder tomar acciones o decisiones sobre el mismo.

Por estos motivos se apunta a analizar solamente el tamaño de los primeros segmentos intercambiados por un flujo lo cual presenta la ventaja adicional que los datos requeridos para este análisis se obtienen a partir de campos disponibles en los encabezados de IP (*Internet Header Length* y *Total Length*) y TCP o UDP (*Header Length* o *UDP Length* respectivamente), no necesitándose por tanto el acceso a los datos de capa de aplicación transportados por el flujo.

3.3. SVM para clasificación de tráfico

La técnica SVM se ha propuesto como técnica de clasificación en varios ámbitos, principalmente en el área de reconocimiento de patrones pero en el área de clasificación de tráfico aparecen también algunas propuestas en la literatura de los últimos años.

SVM se ha propuesto como técnica de clasificación y es utilizada también para estimación por regresión, destacándose por su alto desempeño tanto en la precisión de sus resultados como en su baja complejidad computacional [37, 38].

En el área de clasificación de tráfico, hay varios trabajos recientes que proponen la utilización de SVM con diferentes enfoques y analizando diferentes propiedades de los flujos.

En [38], se propone la utilización de SVM para identificar tráfico *peer-*

to-peer analizando el tamaño de los segmentos, la proporción de conexiones hacia diferentes subredes, la proporción de conexiones establecidas satisfactoriamente y la relación de direcciones IP y puertos usados a nivel de un equipo. En [34] y [35] se analizan múltiples características de los flujos y se aplica SVM, analizándose la influencia de las características en la precisión obtenida en la clasificación. Ambos trabajos concluyen que considerando algunas pocas características se obtienen niveles altos de precisión, por lo que la mayor parte de las características analizadas puede considerarse que no aportan sustancialmente a la mejora de la precisión de la clasificación. En [39] se analizan 10 características de flujos largos correspondientes a tráfico de aplicaciones *peer-to-peer* mediante técnicas combinadas de SVM de una clase y SVM para múltiples clases, obteniéndose buenos resultados de precisión y proponiéndose un método para identificar tráfico desconocido. En [40] se analiza la clasificación de tráfico *peer-to-peer* teniendo en cuenta 12 propiedades de los flujos y aplicando la técnica de SVM, presentándose una alta precisión en la diferenciación de ambos tipos de tráfico. En [4, 41] se propone la técnica de SVM para la identificación de tráfico *peer-to-peer* utilizando la cantidad de segmentos intercambiados en la conexión como característica determinante de los flujos destacándose los resultados de precisión logrados en la clasificación de tráfico de P2P-TV[1]. En [42] se propone la utilización de SVM para diferenciar tráfico *peer-to-peer* y *no peer-to-peer* utilizando una métrica que tiene en cuenta el número de clientes con los que se conecta un cierto equipo, con mejores resultados que los que se obtienen utilizando DPI.

También en el contexto de detección de anomalías, se proponen aplicaciones de SVM. En [43] se utiliza la combinación de SVM para una clase con SVM para múltiples clases, distinguiendo primero el tráfico normal del tráfico anómalo y luego identificando el tipo de anomalía.

En la primera etapa de este trabajo reportada en [1] y en la publicación [36], se propone la utilización de SVM como técnica para la clasificación de tráfico TCP analizando el tamaño de los primeros segmentos de los flujos como característica determinante. Los resultados reportados confirman la aplicabilidad de la técnica de SVM al problema de clasificación de tráfico con la descripción de flujos adoptada.

3.4. *Boosting* de modelos SVM y votación

El análisis de los resultados de precisión de la clasificación mediante la técnica SVM muestra que si bien la precisión global de la clasificación es alta, cuando se analiza la precisión por aplicación, se observan notorias

[1] Peer-to-Peer Streaming

variaciones. Este análisis se presenta en el capítulo 6.

Para mejorar la precisión de la clasificación, en este trabajo se propone la utilización de la técnica de boosting para la generación de una secuencia de modelos SVM mejor adaptados a las diferentes aplicaciones. Esta técnica permite generar secuencialmenete varios modelos SVM de modo que en cada paso el modelo generado mejore la precisión de la clasificación de las muestras mal clasificadas en el paso anterior. Como se detalla en la sección 4.1.2 en esta técnica se trata de mantener constante la precisión global de la clasificación.

La técnica de *boosting* ha sido propuesta para mejorar los clasificadores en otros ámbitos [44, 45, 46]. En el área de las redes de datos, aparecen propuestas para detección de intrusiones en la red en [47].

En este trabajo, se propone como método de clasificación la realización de una votación ponderada de los veredictos de cada modelo SVM generado con la técnica de *boosting*. La ponderación utilizada en este trabajo es la confiabilidad del veredicto de cada modelo para cada aplicación o sea el valor del indicador de "verdaderos positivos". Con este mecanismo de votación se logró mejorar los resultados de precisión por aplicación sin afectar la precisión global.

Los detalles del procedimiento aplicado se incluyen en el capítulo 5.

4 Herramientas estadísticas utilizadas: SVM y *Boosting*

4.1. *Support Vector Machines*

Este trabajo propone la utilización de la técnica de aprendizaje estadístico conocida como *Support Vector Machines* (SVM), introducida por Vladimir Vapnik en 1995 [48]. Sin pretender abarcar en profundidad la teoría de esta técnica, en este capítulo se intentan resumir de forma sencilla y coherente sus principales resultados.

4.1.1. Clasificación binaria

Para describir formalmente el problema de clasificación mediante técnicas de aprendizaje estadístico se considera un conjunto de datos empíricos [49]:

$$(x_1, y_1) \dots (x_m, y_m) \in \mathbb{X} \times \{\pm 1\} \tag{4.1}$$

donde $\mathbb{X}$ es un conjunto de patrones (muestras o entradas) x_i con sus etiquetas (o salidas) asociadas y_i. En este caso se consideran solamente dos clases que para simplificar se etiquetan como $+1$ o -1, lo que corresponde al caso de reconocimiento de patrones o clasificación binaria. Posteriormente se reseñará el caso de múltiples clases.

En el aprendizaje lo que interesa es, dada una nueva muestra $x \in \mathbb{X}$, predecir su correspondiente etiqueta asociada $y \in \{\pm 1\}$. Simplificadamente, esto significa elegir un y de modo que la pareja (x, y) sea similar en algún sentido a las parejas (x, y) utilizadas para entrenar. Esto lleva a una definición de similitud o distancia entre las muestras de $\mathbb{X}$, ya que la similitud en las salidas posibles $\{\pm 1\}$ es simple.

Para medir la similitud se busca una función $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ (denominada *kernel*) que a partir de dos muestras x y x' devuelva un número real que indique la similitud de esas muestras. Una medida de similitud posible

es el *producto interno canónico* entre dos vectores $\mathbf{x}, \mathbf{x'} \in \mathbb{R}^N$, definido como: $(\mathbf{x} \cdot \mathbf{x'}) := \sum_{i=1}^{N} x_i x'_i$, donde x_i es la i-ésima coordenada del vector $\mathbf{x}$. El producto interno permite también calcular la norma o largo de un vector x como: $\| x \| = \sqrt{(\mathbf{x} \cdot \mathbf{x})}$ y se puede definir la distancia entre dos vectores como el largo del vector diferencia entre ambos.

Para usar el producto interno como una medida de similitud, se necesita representar las muestras como vectores en un cierto espacio $\mathbb{H}$ donde sea posible aplicar el producto interno, para lo cual es necesario hacer una transformación: $\Phi : \mathbb{X} \to \mathbb{H}$, que para cada x permita obtener el vector $\mathbf{x} := \Phi(x)$ en el espacio $\mathbb{H}$. Este espacio $\mathbb{H}$ se denomina en la literatura como *feature space*, que podría traducirse como espacio de las propiedades o características.

Esta representación de las muestras o entradas como vectores en el espacio de las propiedades $\mathbb{H}$, permite:

1. Definir la similitud entre dos muestras x y x' como:
$$k(x, x') := (\mathbf{x} \cdot \mathbf{x'}) = (\Phi(x) \cdot \Phi(x')) \tag{4.2}$$
2. Trabajar con las muestras como vectores, usando algebra lineal y geometría analítica

Por otra parte, la transformación Φ podrá seleccionarse de la forma más adecuada al problema a resolver.

En el espacio del producto interno $\mathbb{H}$, se puede utilizar por ejemplo la técnica de clasificación por hiperplanos.

Se considera la clase de hiperplanos
$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \quad con\ \mathbf{w} \in \mathbb{H}, b \in \mathbb{R} \tag{4.3}$$
correspondientes a funciones de decisión de la forma:
$$f(\mathbf{x}) = sgn((\mathbf{w} \cdot \mathbf{x}) + b) \tag{4.4}$$
y se propone un algoritmo de entrenamiento aplicable a los casos en que los datos son separables por hiperplanos, lo que se conoce como *linealmente separables*. De todos los posibles hiperplanos que separan los datos, existe uno que logra el máximo margen de separación entre las m muestras de entrenamiento, esto es:
$$\max_{\mathbf{w},b} \min\{\| \mathbf{x} - \mathbf{x}_i \| \ \ con\ \mathbf{x} \in \mathbb{H}, (\mathbf{w} \cdot \mathbf{x}) + b = 0, i = 1, \ldots, m\} \tag{4.5}$$

Este hiperplano de máximo margen se resuelve como un problema de programación cuadrática, para lo que se conocen algoritmos eficientes [49]. Se puede demostrar, que para encontrar este hiperplano de máximo margen hay que resolver [49]:

$$sujeto\ a: \qquad \begin{cases} \min\limits_{\mathbf{w}\in\mathbb{H},b\in\mathbb{R}} \ \tau(\mathbf{w}) = \dfrac{1}{2}\| \mathbf{w} \|^2 \\ y_i((\mathbf{w}\cdot\mathbf{x}_i)+b) \geq 1, \ i = 1,\ldots,m \end{cases} \tag{4.6}$$

La función objetivo τ, conjuntamente con las restricciones lineales (4.6), conforman un problema de optimización con restricciones tratable con la introducción del *Lagrangiano* y los *multiplicadores de Lagrange*:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\| \mathbf{w} \|^2 - \sum_{i=1}^{m} \alpha_i (y_i((\mathbf{w}\cdot\mathbf{x}_i)+b) - 1) \tag{4.7}$$
$$siendo\ el\ vector\ \boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_m)\ con\ \alpha_i \geq 0$$

Al incluir las restricciones en el Lagrangiano, el problema pasa a ser minimizar $\mathcal{L}$ respecto a las variables $\mathbf{w}$ y b y maximizarlo respecto a las variables α_i. Planteando las condiciones de optimización de Karush-Kuhn-Tucker (KKT) [50], la solución se encuentra cuando se cumple:

$$\frac{\partial}{\partial b}\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = 0, \ \frac{\partial}{\partial \mathbf{w}}\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \tag{4.8}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{4.9}$$

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{4.10}$$

Por tanto el vector solución $\mathbf{w}$, puede expresarse como una combinación lineal de las muestras de entrenamiento, particularmente de aquellas en que el α_i asociado sea no nulo. Estos vectores son los llamados *vectores de soporte* o *support vectors* (SV) que dan nombre a la técnica SVM.

Aplicando las condiciones de KKT, resulta que:

$$\alpha_i[y_i((\mathbf{w}\cdot\mathbf{x}_i)+b) - 1] = 0, \ i = 1,\ldots,m \tag{4.11}$$

y por tanto los SV se ubican en el margen, quedando el hiperplano determinado solamente por las muestras de entrenamiento más cercanas como se muestra esquemáticamente en la figura 4.1.

Sustituyendo 4.9 y 4.10 en 4.7, se eliminan las variables $\mathbf{w}$ y b y se llega al problema de optimización dual, que es el que se resuelve en la práctica.

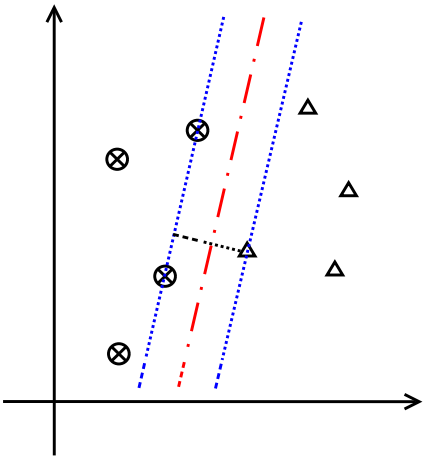

Figura 4.1: Ejemplo de elementos separables.
En rojo (punto-raya) el hiperplano de máximo margen.
En azul (punteado fino) los márgenes determinados por los SV

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4.12}$$

$$sujeto\ a: \quad \begin{cases} \alpha_i \geq 0, \quad i = 1, \ldots, m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

A partir de 4.10, la función de decisión 4.4, puede escribirse como:

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{m} y_i \alpha_i (\mathbf{x} \cdot \mathbf{x}_i) + b \right) \tag{4.13}$$

donde a partir de 4.11 se puede obtener b promediando en todos los valores donde $\alpha_i > 0$ o sea en los SV:

$$b = y_j - \sum_{i=1}^{m} y_i \alpha_i k(w_j, x_i) \tag{4.14}$$

Para aplicar esta técnica, analizada en el espacio $\mathbb{H}$, al espacio de las muestras o entradas, es necesario volver a la ecuación 4.2 que expresa el producto interno a partir de aplicar el *kernel* k a los valores de las entradas. Entonces la función de decisión 4.13 expresada en función de las muestras queda:

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{m} y_i \alpha_i (\Phi(x) \cdot \Phi(x_i)) + b \right) = sgn\left(\sum_{i=1}^{m} y_i \alpha_i (k(x, x_i)) + b \right) \tag{4.15}$$

y el problema dual de optimización a resolver 4.12 queda:

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{4.16}$$

$$\textit{sujeto a}: \quad \begin{cases} \alpha_i \geq 0, \quad i = 1, \ldots, m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

La transformación a utilizar no tiene por que ser lineal como se ha mostrado anteriormente, por lo que si se requiere una superficie de decisión más general, se pueden usar *kernels* no lineales de modo que en el espacio transformado $\mathbb{H}$ aún sea posible realizar una separación por hiperplanos. Si bien se puede demostrar, resulta intuitivo ver que al incrementar las dimensiones del espacio aumentan las posibilidades de separación de los datos. La idea es que eligiendo adecuadamente el *kernel* k a usar en la transformación, se logre que en el espacio $\mathbb{H}$ las clases sean separables por un hiperplano, como se muestra esquemáticamente en la figura 4.2.

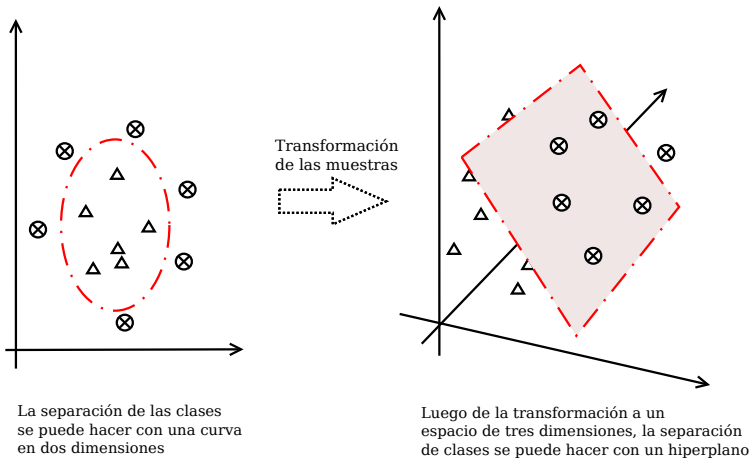

Figura 4.2: La separación de las clases puede ser más sencilla en el espacio $\mathbb{H}$

En muchas situaciones, por ejemplo si las clases se superponen, no tiene por que existir un hiperplano que separe los vectores. Para resolver este problema, se relajan las restricciones incorporando variables de ajuste:

$$\xi_i \geq 0, \quad i = 1, \ldots, m \tag{4.17}$$

de modo que la condición 4.6 se transforma en:

$$y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \tag{4.18}$$

Una posible solución para este clasificador relajado se obtiene de minimizar la función:

$$\tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} \xi_i \tag{4.19}$$

sujeto a las restricciones 4.17 y 4.18 y siendo $C > 0$ una constante que representa el compromiso entre maximizar el margen de separación y minimizar el error de entrenamiento. Usando los multiplicadores de Lagrange como en el caso linealmente separable, se llega al siguiente problema de optimización:

$$\max_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{4.20}$$

$$sujeto\ a: \quad \begin{cases} 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases}$$

La única diferencia con el caso linealmente separable, es que solo importan las muestras cuyo multiplicador α_i sea menor o igual que C, reduciendo el impacto de las muestras "raras" o "*outliers*". La solución es la misma que en el caso anterior y b puede calcularse del mismo modo ya que para los SV con $\alpha_i < C$, el $\xi_i = 0$.

Los *kernels* más usados para efectuar la transformación de las muestras, son:

- Polinómicos de grado d, de la forma:

$$k(x, x_i) = (x \cdot x_i)^d$$

- De base radial gaussiana con parámetro γ:

$$k(x, x_i) = e^{-\gamma \|x - x_i\|^2} \tag{4.21}$$

- Sigmoide de parámetros κ y θ:

$$k(x, x_i) = \tanh(\kappa(x \cdot x_i) + \theta)$$

4.1.2. Clasificación en múltiples clases

En varios escenarios prácticos de clasificación, los datos deben asociarse a más de dos clases. Particularmente en el escenario objeto de este trabajo, interesa diferenciar entre varias clases de tráfico. En esta sección, se mostrarán algunos métodos utilizados para la clasificación en el caso de clases múltiples.

Clasificación uno contra el resto o "*one-versus-all*": Esta técnica consiste en construir un clasificador de M clases a partir de M clasificadores binarios entrenados para separar una clase del resto de las muestras. Una muestra dada, se asociará a la clase para la cual el valor de su correspondiente función de decisión sea máxima.

Clasificación por parejas o "*one-versus-one*": Esta técnica implica la construcción de clasificadores binarios entrenados con parejas de clases, por lo cual se requieren $M(M-1)/2$ clasificadores. Si bien se requieren más clasificadores que en el caso anterior, cada uno debe ser entrenado con menos muestras. En este caso la decisión de clasificación de una nueva muestra se tomará sobre la base de una votación de los clasificadores binarios.

Existen otros métodos propuestos para el caso de clases múltiples pero en base a la relación precisión/complejidad, las implementaciones de SVM utilizan preferentemente el método de clasificación por parejas o "*one-versus-one*". Particularmente este método es el utilizado en la implementación de *LIBSVM* [51] utilizada en este trabajo.

4.2. *Boosting*

La técnica de "*boosting*" surge a partir del algoritmo de entrenamiento llamado *Hedge(β)* propuesto en [52]. En base a lo presentado en [53], si se quiere predecir un cierto evento y se dispone de un conjunto de estrategias posibles (por ejemplo, diferentes estimadores para un cierto parámetro), este algoritmo tiene por objetivo encontrar la mejor estrategia de ese conjunto. Inicialmente se asigna a cada estrategia un peso elegido con una distribución uniforme, ya que a priori no se sabe cuál es la mejor estrategia del conjunto. El procedimiento consiste en ajustar iterativamente los pesos de cada estrategia a medida que se predice un evento (y por tanto se conoce el desempeño de cada estrategia), de modo que las estrategias que lo predicen correctamente incrementan su peso, mientras que aquellas que lo predicen incorrectamente lo disminuyan. El peso de cada estrategia (escalado adecuadamente) puede interpretarse como la probabilidad de que esa estrategia sea la mejor del conjunto.

El esquema de *Hedge(β)* se presenta en el algoritmo 1 y permite obtener el mejor clasificador (aquel con mayor probabilidad p_i^{N+1} al culminar el algoritmo). El algoritmo se parametriza con un valor β menor que 1, que determina cómo se van ajustando los pesos de cada estrategia a medida que se obtiene el resultado de clasificar una nueva muestra con cada una de ellas. Como se puede observar en el esquema, si una estrategia clasifica incorrecta-

mente una muestra ($l_i^j = 1$) su peso se multiplicará por β^1, mientras que si es clasificada correctamente ($l_i^j = 0$) su peso se multiplicará por β^0. Normalizando posteriormente estos pesos, se logra que en cada paso las estrategias que clasifican correctamente un evento aumenten su probabilidad respecto a aquellas que lo predicen incorrectamente.

Si se aplica este algoritmo a la clasificación de muestras, las estrategias de predicción corresponderán a diferentes clasificadores y los eventos serán las muestras a clasificar. A partir de un conjunto de clasificadores que en principio no se sabe cuál es el más apropiado para clasificar una muestra, se elige uno de ellos aleatoriamente con una distribución uniforme. Si este clasificador, clasifica correctamente la muestra, se incrementará su peso para el próximo paso. De este modo en cada paso se elegirá el clasificador que haya logrado mejor precisión en la clasificación de las muestras previas.

Algoritmo 1 Hedge(β)

Entrada: Conjunto D de clasificadores D_i con $i = 1, \cdots, L$
Entrada: Conjunto Z de muestras z_i con $i = 1, \cdots, N$
Tomar $\beta \in [0, 1]$
Asignar pesos iniciales $w^1 = [w_1^1, \cdots, w_L^1]$ con $w_i^1 \in [0, 1]$ y tal que $\sum_{i=1}^{L} w_i^1 = 1$
(Típicamente se toma $w_i^1 = 1/L$)
Inicializar el error acumulado: $\Lambda = 0$
Inicializar el error por clasificador: $\lambda_i = 0$ con $i = 1, \cdots, L$
para cada z_j con $j = 1$ **hasta** N **hacer**
Calcular la distribución: $p_i^j = \dfrac{w_i^j}{\sum_{k=1}^{L} w_k^j = 1}, i = 1, \cdots, L$
para $i = 1$ **hasta** L **hacer**
Calcular el error l_i^j de cada clasificador D_i de modo que

$$l_i^j = \begin{cases} 1 & si\ D_i\ clasifica\ incorrectamente\ a\ z_j \\ 0 & si\ D_i\ clasifica\ correctamente\ a\ z_j \end{cases}$$

fin para
Actualizar el error acumulado: $\Lambda \leftarrow \Lambda + \sum_{i=1}^{L} p_i^j l_i^j$
Actualizar el error por clasificador: $\lambda_i \leftarrow \lambda_i + l_i^j$
Actualizar los pesos $w_i^{j+1} = w_i^j \beta^{l_i^j}$
fin para
Salida: $\Lambda, \lambda_i, p_i^{N+1}$ para $i = 1, \cdots, L$

En [52] se muestra que eligiendo adecuadamente el parámetro β, el error acumulado puede acotarse superiormente a un valor que se aproxima al del mejor clasificador.

Basándose en el algoritmo presentado, en [52] se propone el mecanismo de *boosting* que tiene por objetivo la generación de un predictor muy preciso a partir de la combinación de un conjunto de predictores pobres o de baja precisión. En lugar de partir de un conjunto de clasificadores dado, se trata de construir ese conjunto iterativamente de modo que el clasificador que se agrega en un paso, sea mejor que el clasificador del paso previo. Para lograr

esto, la técnica de *boosting* propone entrenar el clasificador del paso $k+1$ a partir de una selección del conjunto de muestras Z elegidas de modo de preferir aquellas muestras que resultaron mal clasificadas en el paso k. Esto determinará que el clasificador del paso $k+1$ estará más adaptado a las muestras que fueron más difíciles de clasificar correctamente en el paso k. El algoritmo propuesto en [52] es el llamado "*AdaBoost*" (*Adaptative Boosting*) y puede verse como el problema traspuesto al que se plantea en *Hedge(β)*, intercambiando muestras por clasificadores.

En este trabajo se utilizó una implementación del algoritmo *AdaBoost* (detallado en el capítulo 5) para generar por *boosting* una secuencia de clasificadores SVM y posteriormente implementar un método de votación a partir de los veredictos de cada uno de ellos.

5 Metodología e implementación

La metodología utilizada en este trabajo consiste en implementar la identificación de aplicaciones utilizando técnicas de aprendizaje estadístico supervisado. Para validar los métodos de identificación analizados, se requiere contar con tráfico previamente identificado que constituirá la base de la comparación.

A partir de trazas de tráfico identificadas, se construye un conjunto de entrenamiento tomando muestras de todos los tipos de tráfico conocidos. Con este conjunto de entrenamiento se genera un modelo que describe las diferentes clases. El resto de las muestras (las que no se usaron para entrenar) se utilizan para verificar el modelo generado. Comparando la identificación de base de las muestras de prueba, con la identificación obtenida con cada método de clasificación, se puede determinar la precisión de la clasificación.

Como se mencionó anteriormente, se utiliza como descripción de los flujos el tamaño de los primeros segmentos intercambiados, representados de acuerdo a lo detallado más adelante en 5.1.3.

Para que los resultados de los métodos de clasificación analizados tengan una mayor validez, se aplicaron a varias trazas de tráfico, cuyas características se detallan en 5.1.1.

Como primer aproximación al problema se implementó un método simple de clasificación, utilizado en varios trabajos reseñados en la revisión bibliográfica. El método, descrito con más detalle en 5.2.1, consiste en caracterizar cada clase de tráfico por su baricentro y asignar un flujo desconocido a la clase de tráfico cuyo baricentro sea el más cercano. Este método de clasificación permite disponer de valores de base de la precisión de la clasificación que serán comparados posteriormente con los resultados de los métodos propuestos.

El primer método propuesto en este trabajo consiste en utilizar la técnica SVM de acuerdo al esquema detallado en 5.2.2. En el capítulo 6 se presentan los resultados de desempeño de este método en las trazas de tráfico

utilizadas.

La segunda propuesta consiste en utilizar la técnica de *boosting* para generar un conjunto de clasificadores SVM y clasificar el tráfico mediante un sistema de votación de los veredictos de cada uno de los clasificadores. Con este método se logra mejorar aún más la clasificación lograda utilizando un sólo modelo SVM.

Los métodos de clasificación de tráfico utilizados en este trabajo se implementaron a través de programas en lenguaje *perl* [54], manejando la información de los flujos en archivos de texto. Estos programas implementan los algoritmos detallados más adelante en este capítulo y los procesos de automatización de las pruebas de precisión de los mismos. Particularmente la implementación permite el análisis de diferentes trazas de tráfico y generan los datos para el posterior análisis en función de los parámetros relevantes que se indican a continuación:

- cantidad de segmentos utilizados para representar cada cada flujo
- proporción de flujos de entrenamiento y flujos de verificación (para clasificación por distancia a baricentros y SVM)
- valores de C y γ que parametrizan el modelo SVM

Inicialmente se utilizó un computador personal de escritorio para el desarrollo de este trabajo pero posteriormente se implementó todo el sistema de generación de modelos de clasificación y verificación en el Cluster de la Facultad de Ingeniería[1] lo que permitió disminuir sustancialmente el tiempo dedicado a las pruebas. Los programas inicialmente desarrollados debieron ser adaptados al gestor de procesos del Cluster para utilizar varios núcleos de procesamiento simultáneos y ejecutar múltiples entrenamientos de modelos SVM con diferentes parámetros.

En el resto de este capítulo se describen en detalle los diferentes aspectos de la metodología empleada.

5.1. Formato de los datos de prueba

5.1.1. Trazas de tráfico utilizadas

En este trabajo se analizaron tres conjuntos de trazas de tráfico:

[1] http://www.fing.edu.uy/cluster

- Trazas proporcionadas por "The Telecommunication Networks group @ UniBS" (identificadas como *unibs* en el resto del trabajo) [55].
- Trazas puestas a disposición por los autores de [56] (identificadas como *measurement*).
- Trazas de trafico residencial anonimizado provistas por el operador público ANTEL[2] (identificadas como *isp*).

La composición por tipo de aplicación de cada traza se muestra en las tablas 5.1 a 5.6.

Aplicación	**Cantidad de Flujos**
ares	56
bittorrent	284
edonkey	3822
finger	433
http	91
kugoo	59
msnmessenger	44
ntp	369
pop3	965
rtp	119
smtp	746
ssl	281
whois	372
unknown	6117
Total	13758

Tabla 5.1: Composición de flujos TCP de la traza *isp*

Aplicación	**Cantidad de Flujos**
bittorrent	34
dns	141
edonkey	47
nbns	53
tsp	33
unknown	1521
Total	1829

Tabla 5.2: Composición de flujos UDP de la traza *isp*

Aplicación	**Cantidad de Flujos**
bittorrent	1196
emule	2128
http	5162
limeware	80
pop3	646
Total	9212

Tabla 5.3: Composición de flujos TCP de la traza *measurement*

[2]Administración Nacional de Telecomunicaciones

Aplicación	Cantidad de Flujos
bittorrent	34489
emule	2119
limeware	1021
Total	37629

Tabla 5.4: Composición de flujos UDP de la traza *measurement*

Aplicación	Cantidad de Flujos
bittorrent	3697
edonkey	2246
http	27711
imap	47
pop3	465
smtp	75
ssl	3228
unknown	7215
Total	44684

Tabla 5.5: Composición de flujos TCP de la traza *unibs*

Aplicación	Cantidad de Flujos
bittorrent	151
edonkey	44
unknown	112
Total	307

Tabla 5.6: Composición de flujos UDP de la traza *unibs*

Las trazas analizadas consisten en archivos de tráfico capturado en formato $PCAP$[3] por algún mecanismo como "*tcpdump*" [57] o similar. Los flujos indicados como *unknown* en las tablas precedentes, corresponden a tráfico que no pudo ser clasificado por los mecanismos empleados para identificar las demás aplicaciones.

5.1.2. Clasificación de base o "*ground truth*"

Para el análisis de las técnicas de identificación de tráfico con la metodología propuesta, es necesario disponer de trazas de tráfico clasificado. Esto se conoce como "*ground truth*" y es imprescindible para evaluar el desempeño de los métodos de identificación propuestos. Varios autores han propuesto arquitecturas para identificar el tráfico de modo de disponer de trazas clasificadas [58, 59, 60, 61, 62] y algunos señalan incluso que la investigación en el área de clasificación de tráfico no evoluciona más rápidamente porque los métodos que se proponen en los diferentes trabajos, no pueden compararse entre sí por realizar sus evaluaciones sobre diferentes trazas de tráfico.

[3] *packet capture*

Los conjuntos de datos *unibs* y *measurement*, además del archivo de captura en formato *PCAP* traen asociado un archivo que indica para cada flujo de la captura qué protocolo de capa de aplicación transporta ("*ground truth*"). En algunos casos se indica el protocolo y en otros la aplicación que lo generó, a partir de la que se puede inferir el protocolo utilizado. Estas clasificaciones de base se obtuvieron conjuntamente con las trazas y su identificación fue realizada por alguna combinación de los métodos tradicionales como análisis de los puertos de transporte utilizados, análisis de la carga útil de los segmentos transportados o análisis manual.

En el caso del conjunto de datos *isp* fue necesario implementar un sistema para clasificar el tráfico y así obtener la identificación de base. Se optó por realizar la clasificación de base utilizando un sistema que integra el módulo *L7-Filter* [63] para Linux. *L7-Filter* es un mecanismo de clasificación de tráfico que utiliza la técnica de inspección del contenido de los segmentos, realizando una búsqueda de patrones característicos de cada aplicación en los primeros bytes intercambiados por un flujo. Los detalles del sistema implementado se incluyen a continuación.

Implementación de clasificación de base con *L7-Filter*

Para obtener la clasificación de base en el caso del tráfico residencial, se utilizó la técnica de inspección por búsqueda de patrones en la carga útil de los flujos. Esta traza contiene flujos obtenidos mediante una captura de paquetes limitando el tamaño de los mismos a 200 bytes. Por la arquitectura de la red en que se realizó la captura, cada unidad de datos contiene una jerarquía de encabezados anidados: Ethernet, VLAN[4], PPPoE[5], PPP[6], IP, TCP/UDP y el protocolo de capa de aplicación correspondiente (HTTP, SMTP, POP3[7], etc). Por tanto, la carga útil resultante, en el caso que la aplicación se transporte sobre TCP, es de 130 bytes. Este valor surge de restar a los 200 bytes capturados todos los largos de los encabezados de los protocolos mencionados. Esta cantidad de bytes puede ser escasa para el análisis basado en búsqueda de patrones. Esta limitación de los datos disponibles, podría afectar la clasificación de base para los flujos de esta traza pero no afecta los métodos de clasificación propuestos ya que los tamaños de los segmentos para el análisis se obtienen a partir de los encabezados de TCP/UDP/IP.

La identificación de flujos a partir de una traza de tráfico, se realizó en-

[4] *Virtual LAN*
[5] *PPP over ethernet*
[6] *Point to Point Protocol*
[7] *Post Office Protocol 3*

viando la misma por la red a una máquina configurada con *L7-Filter*. Utilizando *L7-Filter* en conjunto con *iptables* [64], es posible obtener la información de la identificación de cada flujo de tráfico en base a la búsqueda de patrones implementada por dicho mecanismo. Para reenviar la traza se utilizaron las herramientas *bittwist* y *bittwisted* [65]. Esta última es un editor de archivos en formato *PCAP* que fue utilizado para quitar los encabezados de VLAN, PPP y PPPoE que dificultaban la identificación por parte de *L7-Filter* mientras que *bittwist* es la herramienta que permite enviar una traza de tráfico a otro equipo.

Con estas herramientas se logró disponer de una correspondencia entre los flujos identificados por sus parámetros característicos (ver sección 2.1) y la aplicación que transportan, obteniendo la información de *ground truth* para la traza *isp*.

5.1.3. Representación de los flujos

La representación utilizada para los flujos en este trabajo está dada por un vector de tamaños de segmentos y una etiqueta que identifica la clase de tráfico o aplicación identificada previamente para ese flujo.

Solamente se consideran en este vector los segmentos cuya carga útil es no nula. Por lo tanto segmentos de establecimiento de conexión en TCP o reconocimientos sin carga útil, no son tenidos en cuenta. La justificación para no tener en cuenta los segmentos con carga útil nula es que estos segmentos son generados por la capa de transporte en base a sus políticas de transmisión, temporizadores y reconocimientos, pero no son generados por la aplicación y por tanto no están relacionados con el comportamiento o las características de la misma.

Además del tamaño de los segmentos intercambiados en un flujo, se considera la secuencia en que fueron intercambiados, agregando un signo negativo en los tamaños de los segmentos que viajan en el sentido reverso del flujo. Se considera que el sentido directo del flujo es aquel cuyo origen es el equipo que inició el establecimiento de conexión en TCP o aquel que envió el primer segmento UDP hacia el destinatario. Esta secuencia de tamaños de segmentos con signo, representa el comportamiento de la aplicación ya que describe el intercambio realizado entre las entidades de capa de transporte que participan del diálogo.

Por ejemplo, para un flujo TCP típico como el esquematizado en la figura 5.1, la representación sería:

$$V^{etiqueta} = (x_1, -y_1, x_2, x_3, x_4, -y_2, x_5)$$

Para obtener esta representación de los flujos es necesario contar para cada flujo con los tamaños de los segmentos intercambiados y la identificación del tipo de aplicación que transporta. Para esto se implementó un programa en lenguaje *perl* que procesa cada archivo de captura, identifica los flujos (a partir de los parámetros clásicos indicados en 2.1) y extrae de la información de los encabezados el tamaño de la carga útil transportada por cada segmento. Vinculando la información de tamaño de los segmentos de cada flujo con la información de la aplicación transportada por ese flujo se obtienen los datos necesarios para aplicar los mecanismos de clasificación propuestos. Este proceso asume que los paquetes en el archivo de captura se encuentran ordenados en el tiempo, lo que es habitual a menos de algún porcentaje despreciable de pérdidas y retransmisiones que pudieran provocar datos desordenados en la traza de tráfico.

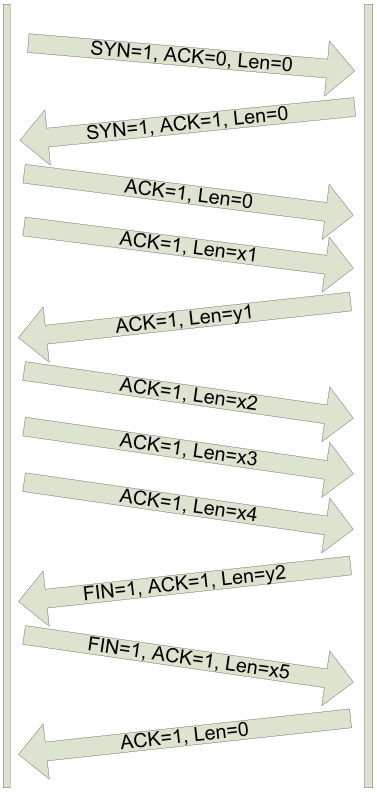

Figura 5.1: Ejemplo de intercambio en un flujo TCP típico

5.2. Métodos de clasificación implementados

5.2.1. Clasificación por distancia a baricentros

Un mecanismo sugerido en varios trabajos para la clasificación de aplicaciones teniendo en cuenta el tamaño de los segmentos intercambiados, consiste en caracterizar cada clase de tráfico por el baricentro del tamaño de sus segmentos. Para esto, se calculan con parte de las muestras el vector de baricentros y se clasifica con un criterio de distancia el resto de la muestra.

El método de clasificación por distancia a baricentros es bastante simple y se utilizó como una primera aproximación al problema, a efectos de obtener los primeros resultados de precisión y para disponer de una base con la que comparar las técnicas propuestas.

Como se mencionó anteriormente, un flujo j de la clase de tráfico m será representado por un vector V_j^m de tamaños de segmentos, donde los

valores positivos representan los segmentos que van desde el originador al destinatario de la comunicación y los negativos aquellos enviados en el sentido opuesto. La representación a utilizar es:

$$V_j^m = (v_{j1}^m, v_{j2}^m, .., v_{jN}^m)$$

donde:

- m es el tipo de tráfico asociado al flujo en la clasificación de base,
- N es el número de segmentos considerados para caracterizar ese flujo (los primeros N intercambiados),
- v_i^m es el tamaño (con signo) del i-ésimo segmento intercambiado por el flujo

En la fase de entrenamiento, se consideran t^m vectores V_j^m para definir el vector de baricentros que caracterizará la clase de tráfico m. Para entrenar cada clase de tráfico se tomó un porcentaje de los flujos disponibles de esa clase. Esta forma de elegir la muestra de entrenamiento permite obtener mejores resultados en la precisión de la clasificación que los reportados en el trabajo previo [1], en el cual se tomaba la misma cantidad de flujos de entrenamiento para todas las clases, independiente de la cantidad de flujos disponibles por clase.

El baricentro de la clase m será un vector $C^m(t^m)$ definido como:

$$C^m(t^m) = (c_1^m(t^m), c_2^m(t^m), .., c_N^m(t^m))$$

donde:

$$c_i^m(t^m) = \frac{\sum_{j=1}^{t^m} v_{ji}^m}{t^m}, \forall i = 1 \ldots N$$

Dado que la carga útil de los segmentos de transporte varía típicamente en una red Ethernet entre 0 y 1460 bytes (teniendo en cuenta los tamaños típicos de encabezados de TCP/UDP/IP), se consideró la desviación estándar para el cálculo de la distancia al baricentro, de modo de mejorar la precisión de este método dando diferentes pesos a las coordenadas de mayor variación en tamaños de segmentos [1].

La desviación estándar para la clase m se define entonces como un vector $S^m(t^m)$:

$$S^m(t^m) = (s_1^m(t^m), s_2^m(t^m), .., s_N^m(t^m))$$

donde:

$$s_i^m(t^m) = \sqrt{\frac{1}{t^m - 1} \sum_{j=1}^{t^m} \left(v_{ji}^m - c_i^m(t^m)\right)^2}\ ,\ \forall i = 1 \ldots N$$

Por tanto, cada clase de tráfico m será representada por una pareja de vectores: $C^m(t^m)$ y $S^m(t^m)$.

En la fase de verificación, se toman los vectores no utilizados en la fase de entrenamiento y se asigna cada vector de prueba a la clase de tráfico más cercana teniendo en cuenta un criterio de distancia.

La distancia de un flujo de prueba W^l a la clase de tráfico m caracterizada por la pareja de vectores $C^m(t^m)$ y $S^m(t^m)$, se define como:

$$d^m(N, t^m) = \sqrt{\sum_{i=1}^{N} \left(\frac{w_i^l - c_i^m(t^m)}{s_i^m(t^m)}\right)^2}$$

La metodología seguida para realizar la identificación de tráfico para cada una de las trazas analizadas aplicando la técnica de baricentros, se presenta esquemáticamente en el algoritmo 2.

Considerando los flujos bien clasificados sobre el total de flujos analizados, se calcula la precisión de la clasificación. Se puede determinar la precisión global de la clasificación (para todos los flujos de verificación) y la precisión por tipo de tráfico. En la sección 6.2.1 se analizan los resultados de la precisión de la clasificación por este método, variando el porcentaje de flujos considerados en la fase de entrenamiento así como la influencia de la cantidad de segmentos considerados para representar los flujos.

5.2.2. Clasificación con SVM

Existen varias bibliotecas que implementan la clasificación con la técnica de SVM. En este trabajo se utilizó "LIBSVM – A Library for Support Vector Machines" implementada por Chih-Chung Chang y Chih-Jen Lin [51], en sus versiones ejecutables desarrolladas en lenguaje *C* para el sistema operativo *linux*.

Los *kernels* gaussianos son los aparecen más frecuentemente utilizados en la literatura, por lo que en este trabajo se utilizó un *kernel* como el mencionado en la ecuación 5.1:

$$k(x, x_i) = e^{-\gamma \|x - x_i\|^2} \tag{5.1}$$

Algoritmo 2 Clasificación por baricentros

Entrada: Conjunto de flujos etiquetados F
  Separar los flujos de F por clases de tráfico de base, F_m
  Fase de entrenamiento
  para cada clase de tráfico m **hacer**
    Tomar TR_m flujos del conjunto F_m
    Calcular el vector de baricentros C_m con los flujos del conjunto TR_m
    Calcular el vector de desviaciones estándar S_m con los flujos del conjunto TR_m y los baricentros calculados C_m
  fin para
  Fase de verificación
  para cada clase de tráfico m **hacer**
    Tomar TE_m flujos de la clase F_m (no incluidos en TR_m)
    para cada flujo t de TE_m **hacer**
      para cada clase de tráfico n **hacer**
        Calcular distancia d_n del flujo t a la clase n
      fin para
      Si $d_{\hat{n}}$ es la mínima d_n, asignar el flujo t a la clase $\hat{n}$
      si $m == \hat{n}$ **entonces**
        devolver El flujo j está **bien** clasificado
      si no
        devolver El flujo j está **mal** clasificado
      fin si
    fin para
  fin para

Utilizando ese *kernel*, el problema de optimización planteado en la ecuación 4.20 queda dependiente de los parámetros C y γ, mencionados en el capítulo 4.

La biblioteca LIBSVM permite generar un modelo de clasificador SVM a partir de un conjunto de muestras de entrenamiento que contengan elementos de todas las clases. Posteriormente ese modelo se verifica con el conjunto de muestras de verificación. A partir del resultado de la clasificación de las muestras de verificación, se puede determinar la precisión de la clasificación de este modelo.

LIBSVM provee dos ejecutables básicos:

- "*svm-train*" para generar un modelo a partir de un conjunto de muestras de entrenamiento. Este modelo en el caso del kernel gaussiano dependerá como se mencionó, de los parámetros C y γ.
- "*svm-test*" para clasificar las muestras de verificación en base al modelo generado con "*svm-train*".

A efectos de evitar que las coordenadas de mayor valor predominen sobre las de menor valor en los vectores que representan las muestras, los

autores de LIBSVM recomiendan en [66], la realización de un escalamiento de los vectores. Para esto se incluye en la biblioteca el ejecutable "*svm-scale*" que lleva por defecto los valores de los vectores al rango $[-1, +1]$. Este escalamiento sirve además para evitar problemas durante los cálculos, en particular cuando se usan *kernels* gaussianos. El esquema utilizado se presenta en el algoritmo 3.

Algoritmo 3 Clasificación con SVM

Entrada: Conjunto de flujos etiquetados F
Separar los flujos de F por clases de tráfico de base F_m
Escalar los vectores con "*svm-scale*"
Fase de entrenamiento
Inicializar conjunto de entrenamiento TR
para cada clase de tráfico m **hacer**
  Tomar TR_m flujos del conjunto F_m y agregarlos a TR
fin para
Crear modelo con "*svm-train*" a partir de TR (el modelo depende de C y γ)
Fase de verificación
para cada clase de tráfico n **hacer**
  Tomar TE_m flujos de la clase F_m (no incluidos en TR_m)
  para cada flujo t de TE_m **hacer**
    Clasificarlo con "*svm-test*" y obtener la clase $\hat{n}$ propuesta por el modelo
    si $m == \hat{n}$ **entonces**
      devolver El flujo j está **bien** clasificado
    si no
      devolver El flujo j está **mal** clasificado
    fin si
  fin para
fin para

Para determinar los valores de C y γ que maximicen la precisión del modelo, los autores proponen en [66], la generación de diferentes modelos recorriendo una grilla de valores de C y γ. Posteriormente se elegirá el modelo que maximice la precisión, determinándose entonces los valores adecuados de esos parámetros. En [66] se realizan además recomendaciones del rango a cubrir con los parámetros C y γ así como sus incrementos, sugiríendose el uso de una secuencia exponencialmente creciente de valores, tanto para el parámetro C como para γ. En las primeras etapas de este trabajo y particularmente en lo reportado en [1] se trabajó con una grilla extendida considerando los siguientes valores:

$$C = 2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 1, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}, 2^{15}$$

$$\gamma = 2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 1, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}, 2^{15}$$

Los resultados obtenidos en [1] permitieron acotar los rangos a recorrer y los resultados reportados en este documento se realizaron utilizando los siguientes valores:

$C = 2^{-1}, 1, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}$ y

$\gamma = 2^{-1}, 1, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}$

En la sección 6.2.2 se muestran los resultados de la clasificación con SVM, analizándose la influencia de los parámetros C y γ, así como el efecto de la variación de la cantidad de flujos de entrenamiento y la cantidad de segmentos por flujo.

5.2.3. Clasificación por votación de modelos SVM

La técnica de *boosting* aplicando el algoritmo *AdaBoost* permite generar una secuencia de modelos SVM de forma iterativa, de forma tal que en cada paso, el modelo generado clasifique mejor las muestras mal clasificadas en el paso anterior. Se aspira a que este mecanismo contribuya a aumentar la precisión de las clases de tráfico para las que se obtenga una precisión menor en la clasificación utilizando solamente un modelo SVM.

El esquema utilizado es el que se muestra en el algoritmo 4 y consiste en generar modelos SVM utilizando diferentes conjuntos de flujos de entrenamiento. Inicialmente, de un conjunto de flujos se toman con una distribución uniforme un cierto porcentaje (50 % en este trabajo) para entrenar un modelo SVM. Posteriormente se verifica la precisión global de ese modelo también con una muestra uniforme del mismo porcentaje de flujos tomados del conjunto inicial. En base al resultado de la verificación del modelo, se modifican los pesos de cada flujo aumentando el peso de aquellos que fueron mal clasificados por el modelo. De esta forma, para el siguiente paso, ya no se tomarán flujos de entrenamiento y verificación con una distribución uniforme, sino que tendrán mayor preferencia los flujos mal clasificados en el paso anterior. Este procedimiento, llevará a que los modelos que se vayan generando en la secuencia de iteraciones se ajusten a diferentes tipos de tráfico. El algoritmo *AdaBoost* incluye un método para ajustar los pesos en cada paso de modo de aumentar el peso de los mal clasificados y disminuir el peso de los bien clasificados. Es necesario en cada paso escalar adecuadamente los pesos para mantener su suma en 1. En este trabajo se parametrizó arbitrariamente el algoritmo *AdaBoost* con $K = 10$ generándose 10 modelos SVM. A partir de los modelos SVM generados con este algoritmo debe decidirse cómo clasificar un flujo dado.

La propuesta de este trabajo consiste en tomar como resultado de la clasificación una votación ponderada de los veredictos de los 10 modelos SVM generados por el algoritmo 4. Para la ponderación del veredicto de cada modelo SVM se toma en cuenta la **confiabilidad** de la respuesta de cada

Algoritmo 4 AdaBoost

Entrada: conjunto de muestras de entrenamiento T conteniendo l flujos x_i etiquetados
Asignar pesos iniciales a cada flujo en forma uniforme $p_1(x_i) = 1/l$
para $k = 1$ **hasta** K **hacer**
Construir un conjunto de entrenamiento TR_k tomando m flujos del conjunto T en base a los pesos $p_k(x_i)$
Generar modelos $SVM_k(C, \gamma)$ usando el conjunto TR_k (variando C y γ se generan varios modelos)
Construir un conjunto de prueba TE_k tomando m flujos de T en base a los pesos $p_k(x_i)$
Verificar el modelo $SVM_k(C, \gamma)$ usando el conjunto de flujos TE_k
(se prueban todos los modelos generados por variación de C y γ)
Elegir el modelo $SVM_k(C, \gamma)$ que obtenga la máxima precisión global
Calcular el error $e_k = \sum_{i=1}^{m} p_k(i) \mid i\ fue\ mal\ clasificado$
Calcular $\alpha_k = \frac{1}{2} \log \frac{e_k}{1 - e_k}$ (notar que para $e_k < 0{,}5$, $\alpha_k < 0$)
para $j = 1$ **hasta** m **hacer**
$p'_{k+1}(x_j) = p_k(x_j) \times \begin{cases} e^{\alpha_k} & si\ x_j\ fue\ correctamente\ clasificado \\ e^{-\alpha_k} & si\ x_j\ fue\ mal\ clasificado \end{cases}$
fin para
Normalizar los pesos $p_{k+1}(x_j) = \frac{p'_{k+1}(x_j)}{N_k}$, con N_k tal que $\sum_{i=1}^{l} p_{k+1}(x_i) = 1$)
fin para

modelo para esa aplicación o lo que es lo mismo el porcentaje de **Verdaderos Positivos, TP** de ese modelo para esa aplicación.

En este método un flujo será verificado por cada uno de los modelos SVM generados y cada uno de ellos le asignará en una clase. Cada veredicto tendrá asociada una confiabilidad que puede ser calculada en la fase de verificación del modelo cuando se calcula también su precisión. Dicho de otra forma, si un modelo clasifica el tráfico como de la clase X, se considerará la confiabilidad de esa respuesta, teniendo en cuenta qué porcentaje de las veces que el tráfico era realmente de clase X, el modelo respondió acertadamente.

	Modelo SVM #									
	0	1	2	3	4	5	6	7	8	9
Veredicto	http	http	http	ssl	edonkey	edonkey	ssl	http	http	http
Confiabilidad	99.051	97.836	97.562	82.949	59.280	58.699	65.229	98.369	99.968	99.681

Tabla 5.7: Ejemplo de veredictos y sus confiabilidades para los diferentes modelos SVM

A modo de ejemplo, para el tráfico TCP de la traza identificada como *isp* tomando 5 segmentos por flujo y entrenando cada clase con 15 % del tráfico disponible para esa clase se obtienen, para los diferentes modelos SVM generados por el algoritmo *AdaBoost*, diferentes confiabilidades. En las tablas 5.8 y 5.9 se muestran los resultados de confiabilidad para los modelos SVM 0 y 3 de la secuencia. Para un flujo de verificación particular, con clasificación

de base **http**, cada modelo SVM de la secuencia dicta un veredicto y cada dictamen tiene una cierta confiabilidad como se muestra en la tabla 5.7. En este caso, el veredicto final toma en cuenta que se pronunciaron:

- por **http** los modelos 0, 1, 2, 7, 8, 9; sumando una confiabilidad de 592.467
- por **ssl** los modelos 3 y 6; sumando una confiabilidad de 148.178
- por **edonkey** los modelos 4 y 5; sumando una confiabilidad de 117.979

y por tanto el mecanismo de votación propuesto, asignará este flujo a la clase **http**, tomando la respuesta de mayor confiabilidad acumulada.

Los resultados de la precisión de clasificación obtenidos con la aplicación de esta técnica, se analizan en la sección 6.3.2.

5.3. Tráfico desconocido

En el problema de clasificación de tráfico abordado en este trabajo, es recurrente en la literatura el planteo de ¿qué pasa con el tráfico desconocido? [39].

Cuando se realiza el aprendizaje estadístico, se entrena el sistema con muestras previamente clasificadas y en la fase de verificación se intentará asociar las muestras a alguna de las clases definidas en la fase de entrenamiento. Si se pretende utilizar un sistema de este tipo para clasificar tráfico real, seguramente habrá flujos que no pertenecerán a ninguna de las clases definidas en la fase de entrenamiento. Esto puede darse porque se trata de una nueva aplicación o porque ese tipo de tráfico no estaba contemplado en la muestra de entrenamiento.

En la literatura aparecen varias propuestas para distinguir cuándo una muestra no pertenece a ninguna de las clases conocidas y por tanto debe considerarse desconocida. Típicamente se plantea la definición de algún tipo de distancia que permita tomar esa decisión. En algunos trabajos se plantea la creación de una clase ficticia que que represente al tráfico desconocido, por ejemplo para el caso de analizar el tamaño de los segmentos, se toman muestras aleatorias uniformemente distribuidas en el rango de valores posibles. Este rango queda definido en la mayor parte de los casos por la carga útil de las tramas Ethernet o sea 1500 bytes [36, 41].

En este trabajo no se abordó este problema en los términos descritos anteriormente, pero sí se analizó el comportamiento de la precisión de la clasificación cuando se incluye una clase de tráfico desconocido. Dado que 2 de las trazas disponibles contaban con flujos no clasificados, se analizó el efecto de utilizar esos flujos desconocidos para entrenar una clase asociada a ese tráfico. Es decir que se utilizan muestras de tráfico real para entrenar una clase de tráfico desconocido, por entenderse que la propuesta de considerar una distribución de tamaños de paquetes uniformemente distribuido en el rango $[-1500, +1500]$ no representa correctamente al tráfico real, dado que las tramas Ethernet tienen un tamaño mínimo no nulo.

En el capítulo 6 se analizan los resultados de la precisión en la clasificación considerando o no esta clase.

Aplicación	Veredicto del modelo SVM #0												
	whois	bittorrent	edonkey	finger	http	kugoo	ares	msnmessenger	ntp	pop3	rtp	smtp	ssl
whois	46.952	0.000	3.063	1.887	0.000	0.000	0.000	11.177	0.202	5.961	0.000	24.567	0.000
bittorrent	1.145	88.484	1.004	7.801	0.000	0.000	0.000	0.000	0.265	0.000	0.349	0.000	0.000
edonkey	0.128	0.188	65.054	0.734	0.033	0.669	0.000	0.000	0.393	0.000	0.026	0.000	0.059
finger	2.627	1.663	6.844	0.000	0.578	35.431	0.000	2.802	14.922	0.183	0.458	0.000	0.000
http	0.000	0.000	0.627	19.607	99.051	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
kugoo	1.374	3.043	5.780	34.685	0.000	25.932	0.000	2.929	21.589	0.000	1.675	0.000	0.000
ares	0.000	0.000	3.068	4.845	0.000	0.000	99.184	0.000	2.696	0.000	0.000	0.000	0.000
msnmessenger	43.743	0.000	1.278	6.056	0.000	0.000	0.000	81.627	1.685	0.000	0.000	0.000	0.000
ntp	0.440	1.218	5.398	11.106	0.338	31.828	0.816	0.000	55.909	0.000	1.073	0.000	2.130
pop3	0.000	0.000	0.000	0.168	0.000	0.000	0.000	0.000	0.000	90.674	0.000	1.282	0.000
rtp	0.000	3.803	5.780	8.215	0.000	4.322	0.000	1.465	1.270	0.000	77.042	0.000	6.652
smtp	3.591	0.000	0.076	0.289	0.000	0.000	0.000	0.000	0.000	3.182	0.000	73.907	0.000
ssl	0.000	1.600	2.027	4.608	0.000	1.818	0.000	0.000	1.069	0.000	19.378	0.244	91.159
Total	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000

Tabla 5.8: Confiabilidad del veredicto del modelo SVM #0 para cada aplicación. Traza: *isp*, 5 segmentos por flujo, 15 % de tráfico utilizado para entrenar

Aplicación	Veredicto del modelo SVM #3												
	whois	bittorrent	edonkey	finger	http	kugoo	ares	msnmessenger	ntp	pop3	rtp	smtp	ssl
whois	40.555	0.000	2.071	1.418	0.000	0.828	0.000	13.367	0.399	7.705	0.000	25.412	0.000
bittorrent	0.537	86.485	1.939	3.720	0.000	0.000	0.000	0.000	0.785	0.000	0.963	0.000	0.758
edonkey	0.000	0.154	62.994	0.525	0.033	0.726	0.000	0.048	0.446	0.000	0.071	0.000	0.028
finger	2.288	1.753	7.881	34.621	0.000	33.458	0.211	3.404	12.341	0.179	1.683	0.000	0.000
http	0.000	0.000	2.422	16.261	99.967	3.391	0.000	0.000	0.000	0.000	0.000	0.000	1.183
kugoo	2.577	7.128	6.512	25.874	0.000	31.261	1.545	3.114	18.817	0.000	3.080	0.000	1.817
ares	0.000	0.000	3.950	0.000	0.000	0.000	96.760	0.000	2.663	0.000	0.000	0.000	0.000
msnmessenger	51.285	0.000	0.000	2.368	0.000	0.000	0.000	78.510	1.665	0.000	0.000	0.000	0.000
ntp	0.000	1.141	4.915	8.713	0.000	22.521	1.484	0.000	57.438	0.000	1.973	0.000	1.455
pop3	0.079	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	88.692	0.000	1.351	0.000
rtp	1.932	2.139	4.186	3.123	0.000	7.815	0.000	1.557	4.391	0.000	70.846	0.000	11.811
smtp	0.204	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.424	0.000	73.005	0.000
ssl	0.542	1.200	3.131	3.378	0.000	0.000	0.000	0.000	1.056	0.000	21.383	0.232	82.949
Total	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000	100.000

Tabla 5.9: Confiabilidad del veredicto del modelo SVM #3 para cada aplicación. Traza: *isp*, 5 segmentos por flujo, 15 % de tráfico utilizado para entrenar

6 Análisis de resultados

En este capítulo se muestran y analizan los resultados obtenidos de acuerdo a la metodología detallada en los capítulos previos.

Se realiza un análisis de la precisión de los diferentes métodos de clasificación implementados: clasificación por distancia a baricentros, clasificación con la técnica de SVM y clasificación por votación de clasificadores SVM generados con la técnica de *boosting*. Se presentan los resultados en forma gráfica de modo de poder visualizar el comportamiento de los métodos en función de los varios parámetros.

Los resultados de los métodos de clasificación analizados deben discutirse en función de varios aspectos:

- La cantidad de segmentos de cada flujo considerados para describir el tráfico
- El porcentaje de flujos utilizados para entrenar los modelos
- El protocolo de transporte: TCP o UDP
- La variación de los parámetros C y γ, en el caso de aplicarse la técnica de SVM
- El efecto de la inclusión o no del tráfico desconocido
- La consideración de la precisión global y la precisión por aplicación
- El comportamiento de los diferentes métodos para las diferentes trazas de tráfico disponibles descritas en 5.1.1

En las próximas secciones de este capítulo se analizará la precisión global de la clasificación de cada traza utilizando los diferentes métodos considerados y se discutirá la influencia de los parámetros mencionados en el párrafo anterior. A partir de estos primeros resultados se acotará el análisis a algunos valores de la cantidad de segmentos utilizados para describir los flujos y del porcentaje de flujos utilizados para la fase de entrenamiento.

Definidos así los rangos relevantes de estos parámetros, se analizará la precisión de la clasificación por tipo de aplicación.

Además será relevante considerar el efecto del tráfico desconocido. Como se mencionó en el capítulo 5, en este trabajo se propone la consideración del tráfico desconocido como una clase más, por lo que se analizará la influencia de esta decisión en la precisión por aplicación de los métodos evaluados.

Si bien se realizó un análisis completo de la variación de todos los parámetros para las tres trazas de tráfico disponible, las gráficas que se incluyen en este capítulo se han elegido para ejemplificar y sustentar el análisis. La totalidad de las gráficas se encuentran disponibles en `http://iie.fing.edu.uy/~ggomez/maestria/graficas/`.

6.1. Flujos y Baricentros en tres dimensiones

Antes de comenzar el análisis de los resultados obtenidos, es interesante mostrar una visión en el espacio de las componentes de los vectores que se utilizan para identificar los flujos, es decir, los tamaños de los primeros segmentos intercambiados. A modo de ejemplo, se presentan gráficas de los flujos disponibles para algunos protocolos, restringidos a tres dimensiones. Esto aporta una aproximación intuitiva a cómo se distribuyen en el espacio diferentes flujos identificados con un mismo protocolo cuando se consideran solamente los tres primeros segmentos. Se señala también el baricentro resultante de promediar los tamaños de cada segmento.

Como se aprecia en las gráficas 6.1, 6.2, 6.3 (protocolo de transporte TCP) y 6.4, 6.5, 6.6 (protocolo de transporte UDP), intuitivamente puede verse que algunas aplicaciones presentan patrones espaciales más o menos delineados, lo que permite suponer que hay un comportamiento característico determinado por el tamaño de los primeros segmentos intercambiados (en este caso los primeros 3 segmentos).

También puede observarse que algunas aplicaciones podrían caracterizarse medianamente bien por su baricentro (bittorrent, dns), mientras que otras (http, skype) presentan una dispersión espacial de sus flujos que permiten prever bajos índices de precisión en su clasificación con el método de distancia a baricentros.

Se observa también que los baricentros de algunas aplicaciones se encuentran muy cercanos entre si, lo que permite prever la dificultad distinguir esos flujos con el criterio de distancia a baricentros.

Las gráficas presentadas corresponden a una aplicación de cada protocolo de transporte para cada una de las trazas de tráfico analizadas.

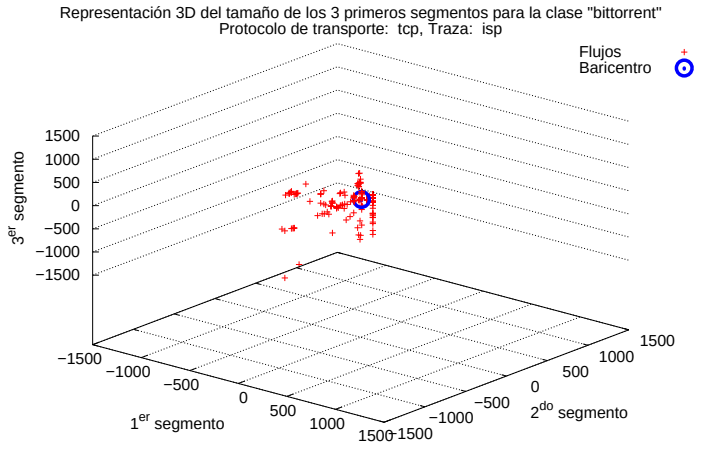

Figura 6.1

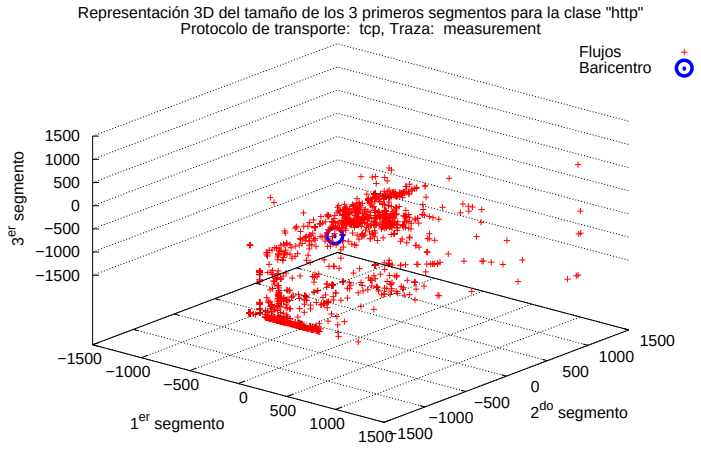

Figura 6.2

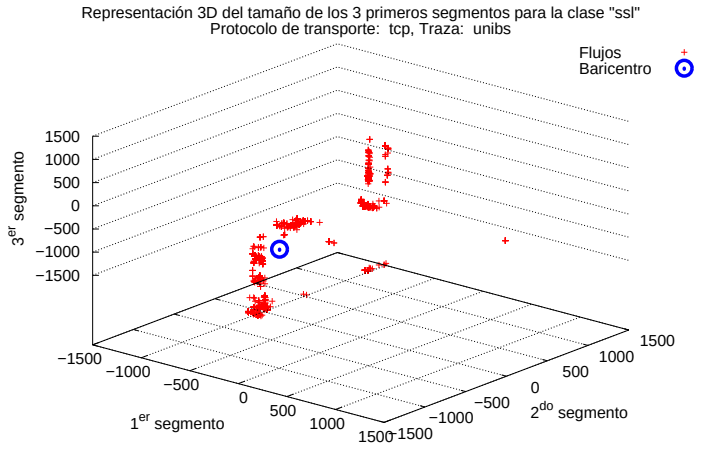

Figura 6.3

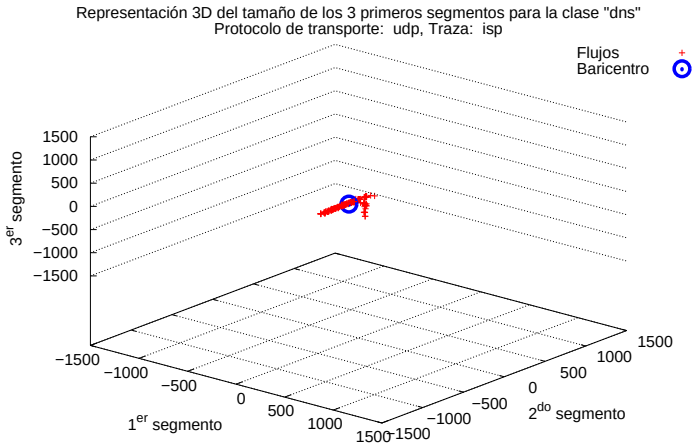

Figura 6.4

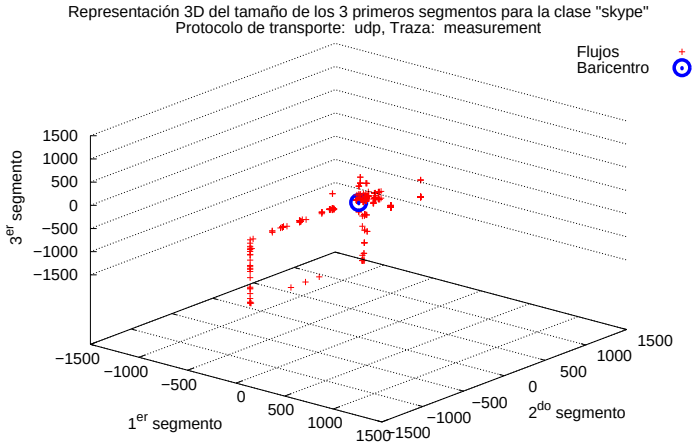

Figura 6.5

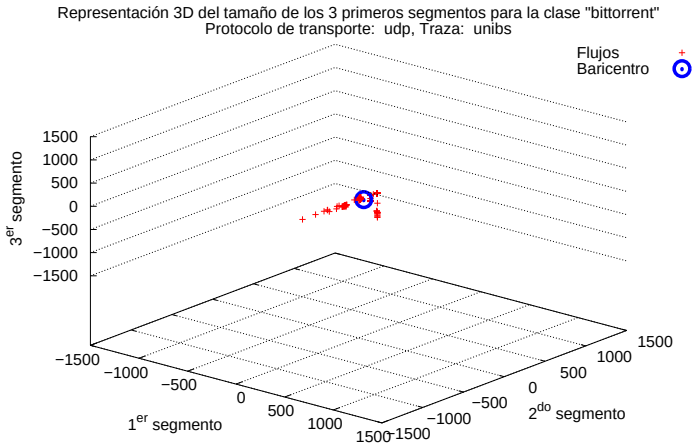

Figura 6.6

6.2. Precisión global de la clasificación

6.2.1. Clasificación por distancia a baricentros

En esta sección se analiza la variación de la precisión global obtenida con la técnica de distancia a baricentros, en función de la cantidad de segmentos utilizados para representar los flujos. Las gráficas presentadas, permiten además analizar la influencia del porcentaje de flujos utilizados en la fase de entrenamiento, en este caso, para calcular los baricentros.

Las gráficas muestran en general un comportamiento bastante plano tanto para TCP (gráficas 6.7, 6.8, 6.9) como para UDP (gráficas 6.10, 6.11, 6.12) en las tres trazas analizadas.

Se puede observar que mayoritariamente se obtienen las mejores precisiones cuando se consideran 2, 3 o 4 segmentos para representar los flujos y que la precisión presenta un comportamiento estable, calculando los baricentros con 30 % o más de flujos por cada tipo de tráfico. Salvo para el tráfico TCP en la traza *unibs*, en las restantes gráficas la precisión global no alcanza al 80 %.

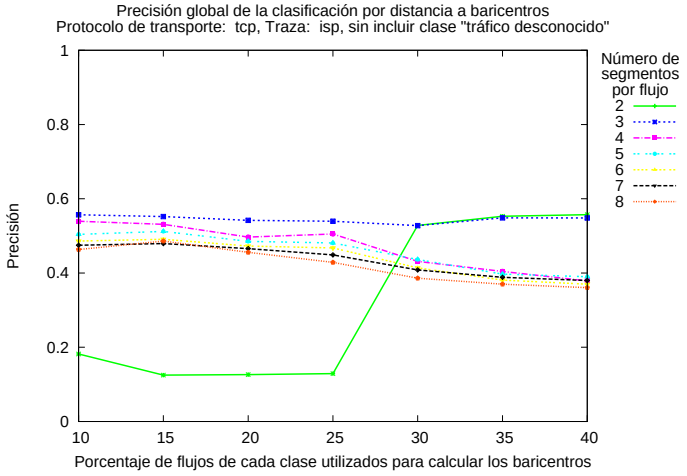

Figura 6.7

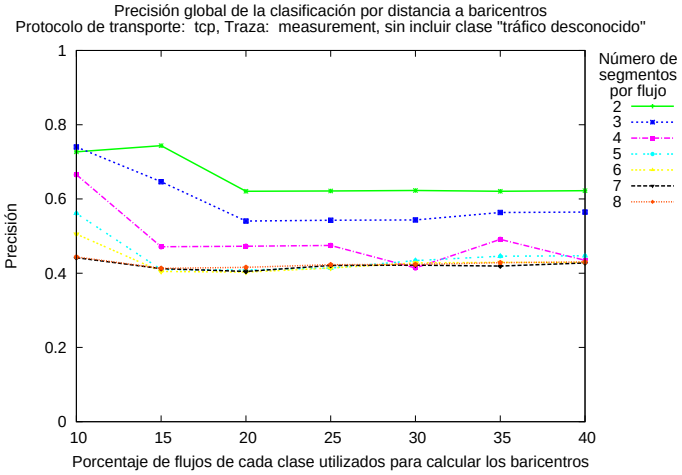

Figura 6.8

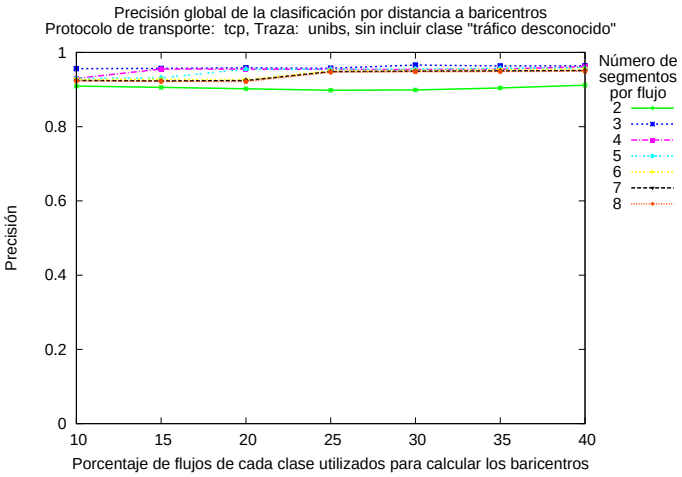

Figura 6.9

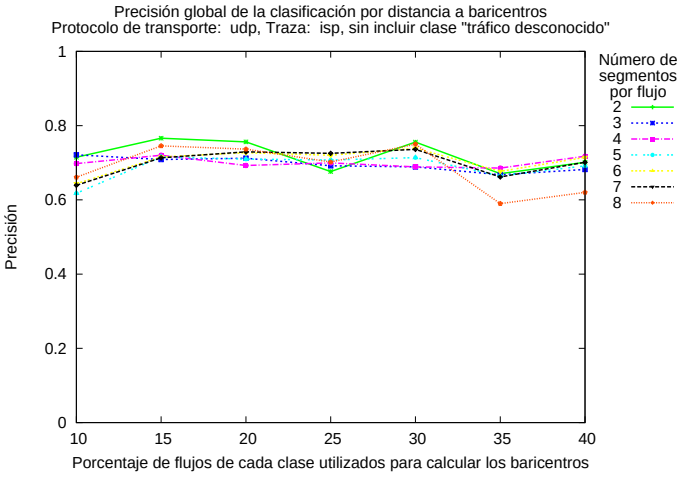

Figura 6.10

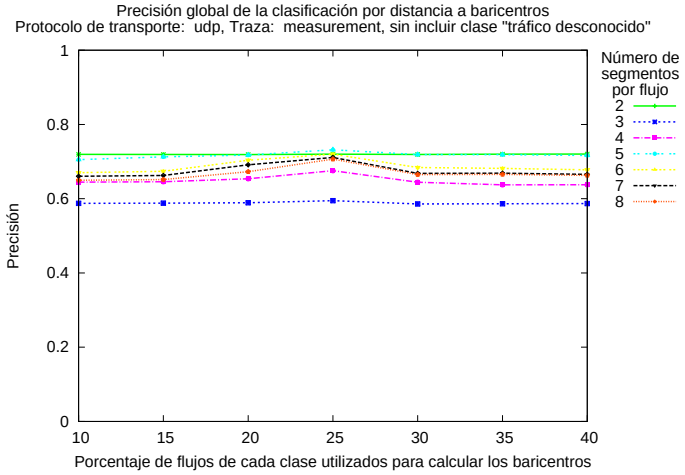

Figura 6.11

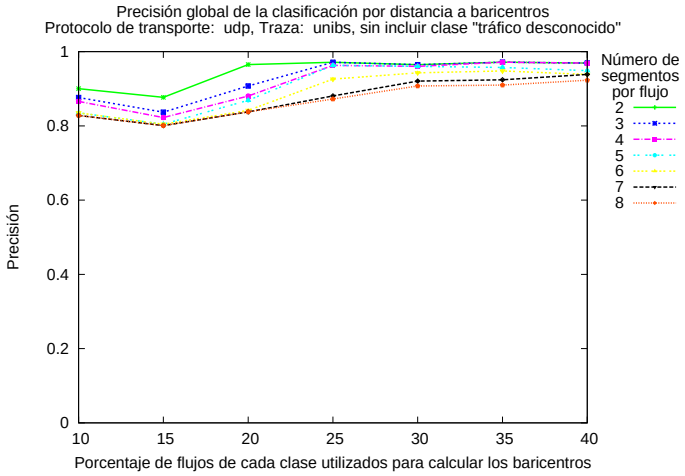

Figura 6.12

6.2.2. Clasificación con SVM

Efecto de la variación de C y γ

En esta metodología, la precisión de los resultados depende de los parámetros C y γ que determinan el modelo SVM. Como se detalló en el capítulo 5, se generó un conjunto de modelos SVM variando esos parámetros dentro de una grilla. Posteriormente se eligió el modelo que obtuviera los mejores resultados de precisión global.

El efecto de los parámetros C y γ en la precisión global de la clasificación se muestra en las gráficas 6.13, 6.14 y 6.15. Los datos se representan en escala logarítmica y muestran leves variaciones de la precisión global dentro de los rangos considerados. Como se mencionó en el capítulo 5, la definición de los límites de esta grilla se basó en los resultados preliminares publicados en [1].

No se observan particularidades destacables en el protocolo de transporte empleado (TCP o UDP) por lo que se seleccionaron solamente tres gráficas para incluir en este capítulo.

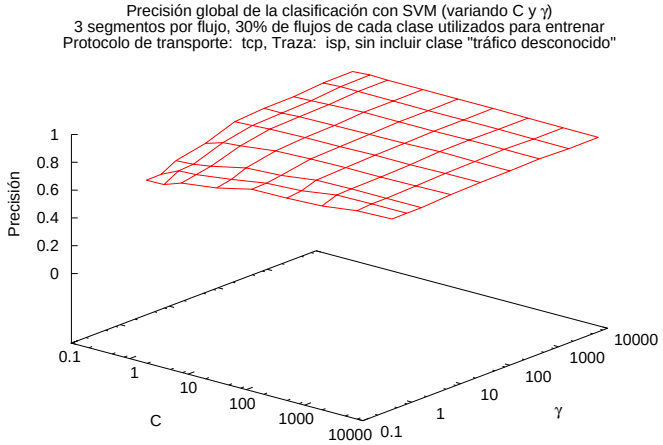

Figura 6.13

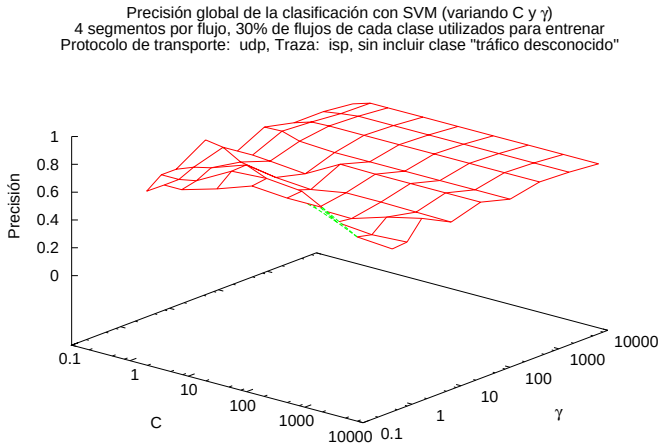

Figura 6.14

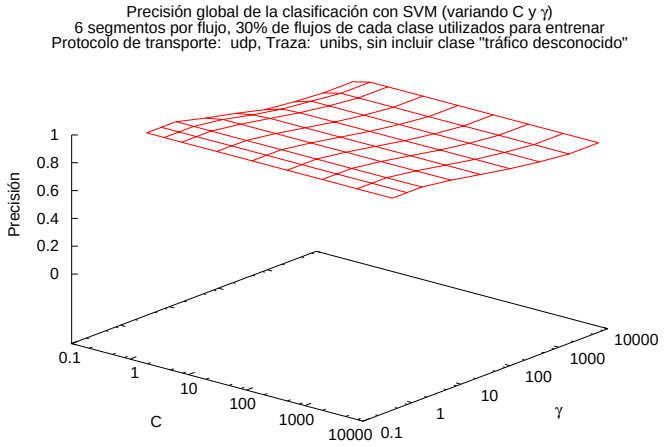

Figura 6.15

Precisión global utilizando SVM

Se presentan en este apartado las gráficas de precisión global de la clasificación con SVM para valores de los parámetros C y γ que maximizan dicha precisión. Las gráficas son paramétricas en función de la cantidad de segmentos considerados para cada flujo así como de la cantidad de flujos utilizados para entrenar el modelo SVM.

En las gráficas 6.16 a 6.21, se pueden observar comportamientos similares a los de la clasificación por distancia a baricentros, en el sentido de la influencia de la cantidad de flujos de entrenamiento y la cantidad de segmentos a considerar para la representación de cada flujo. Los valores de precisión alcanzados son notoriamente superiores a los del método de clasificación por baricentros razón por la que se optó por cambiar el rango del eje de las ordenadas a [0.7,1] para permitir una mejor visualización de las variaciones.

Los resultados indican que la metodología SVM arroja muy buenos resultados de precisión tanto para TCP como para UDP. Los resultados de menor precisión se observan en la traza denominada *isp*, lo que puede deberse a las limitaciones de la traza disponible y su efecto en la clasificación de base realizada (ver sección 5.1.2).

Con este método se puede observar que dependiendo de la traza considerada las mejores precisiones se obtienen nuevamente considerando 2, 3 o 4 segmentos para representar los flujos y en ese rango de valores la precisión presenta un comportamiento estable o creciente con el porcentaje de flujos utilizados en la fase de entrenamiento. Estos valores son consistentes con los resultados obtenidos en la clasificación por distancia a baricentros. Para los menores valores de precisión obtenidos (gráfica 6.19) se observa que a partir de 30 % flujos de entrenamiento ya los valores son razonablemente altos y en todos los casos la precisión mínima en el rango de valores mencionado supera el 80 %, máximo alcanzado por el método de distancia a baricentros.

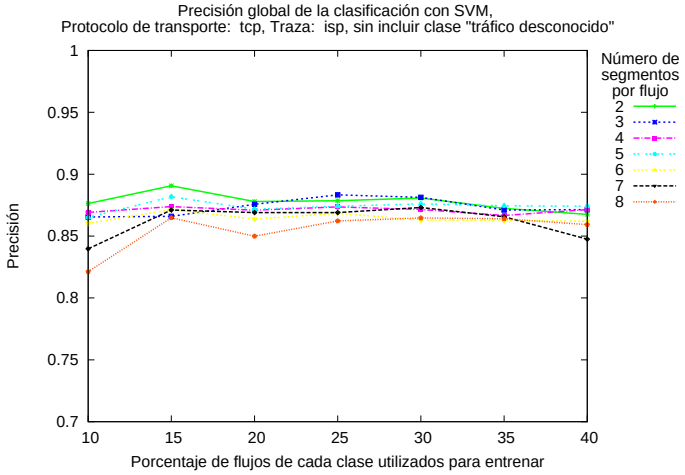

Figura 6.16

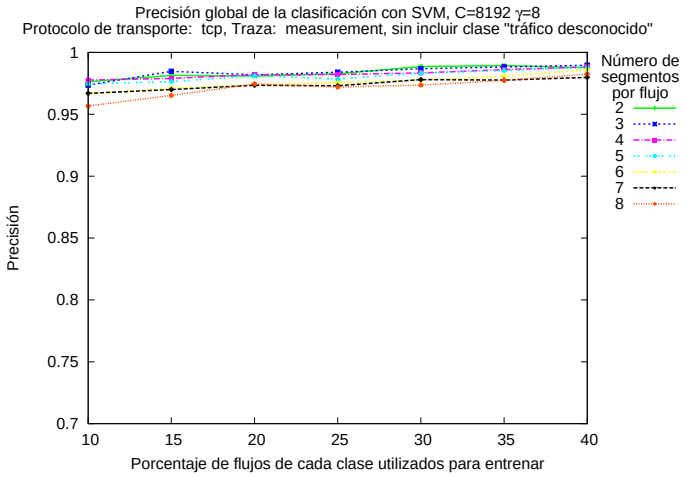

Figura 6.17

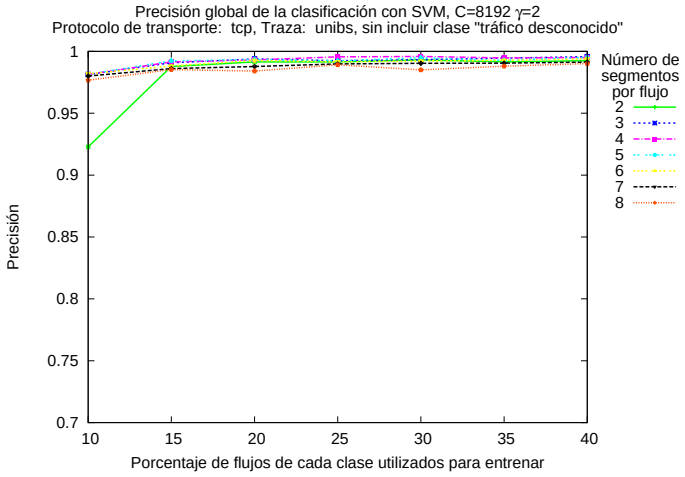

Figura 6.18

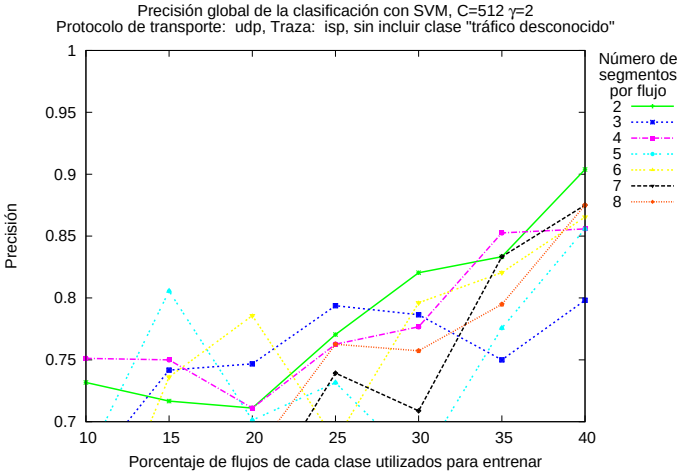

Figura 6.19

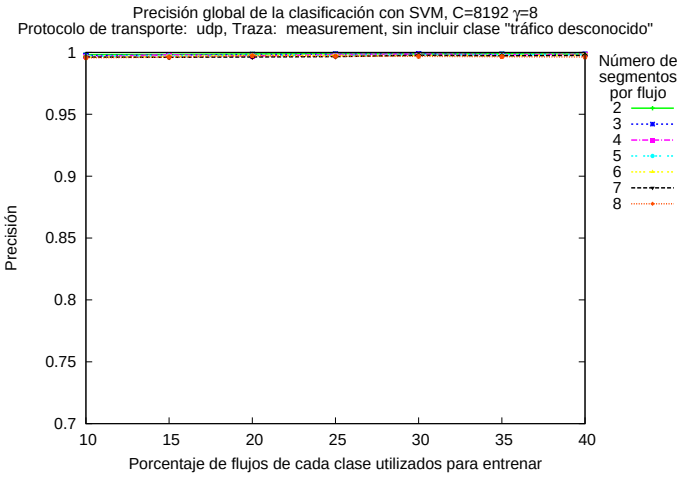

Figura 6.20

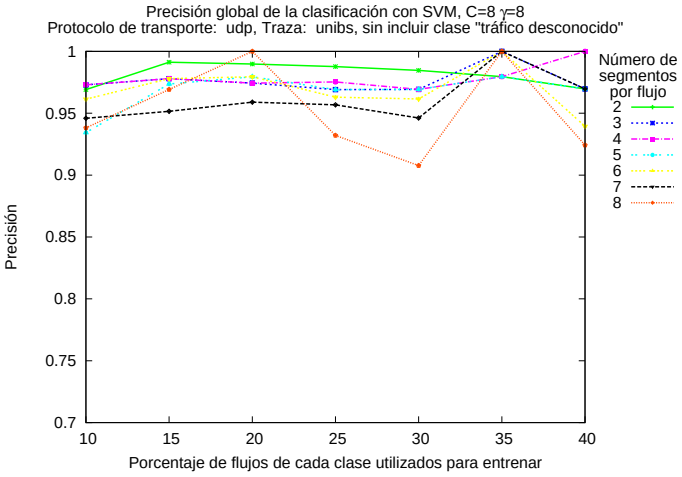

Figura 6.21

6.2.3. Comparación de precisión global

Las gráficas que se presentan a continuación permiten comparar la precisión global obtenida por los dos métodos analizados hasta aquí. En todos los casos puede observarse que el método de clasificación con SVM presenta una precisión superior al método de clasificación por distancia a baricentros. Se escogen a modo de ejemplo algunas gráficas para las tres trazas utilizando 3 y 4 segmentos por flujo, presentándose resultados tanto para protocolo TCP como UDP.

En el Apéndice A se presentan los resultados completos –en el rango de valores analizado– de la clasificación por distancia a baricentros y SVM en forma de tabla, de modo que se pueda consultar algún caso que no aparece en las gráficas presentadas.

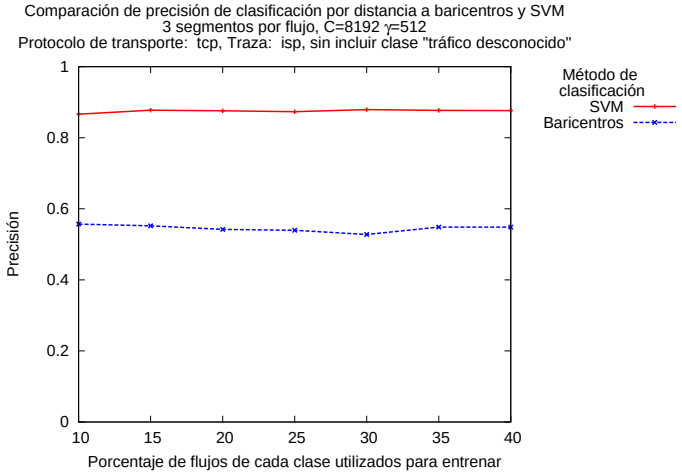

Figura 6.22

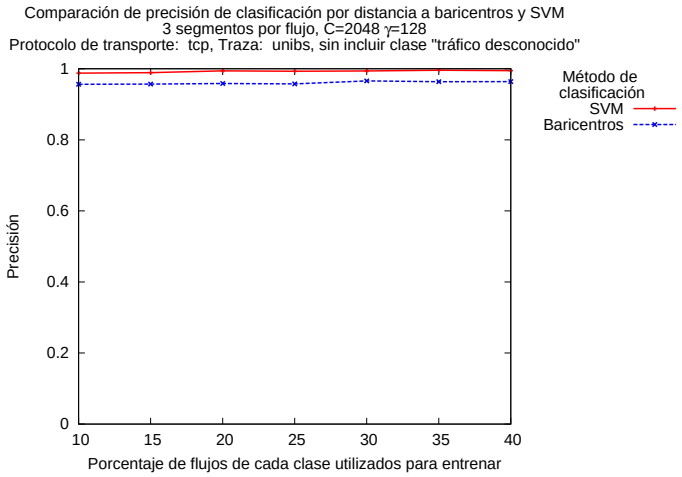

Figura 6.23

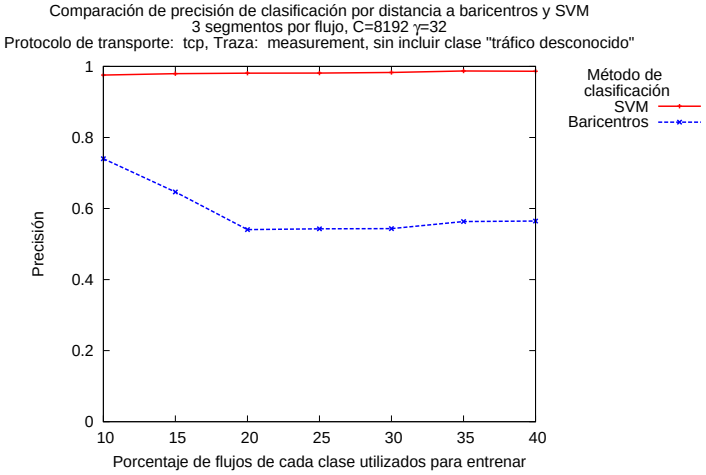

Figura 6.24

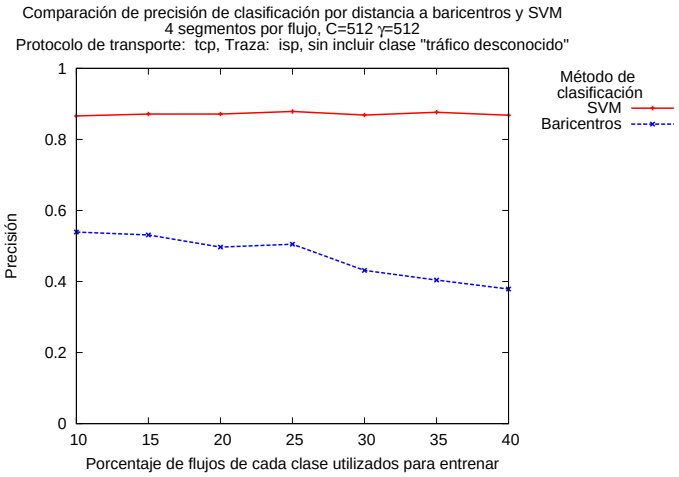

Figura 6.25

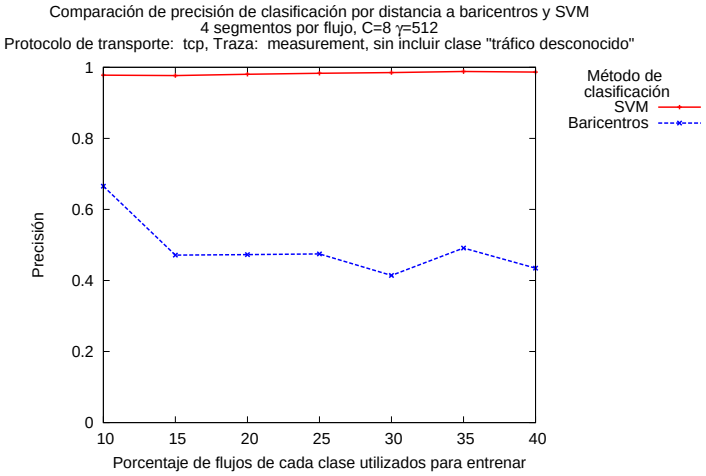

Figura 6.26

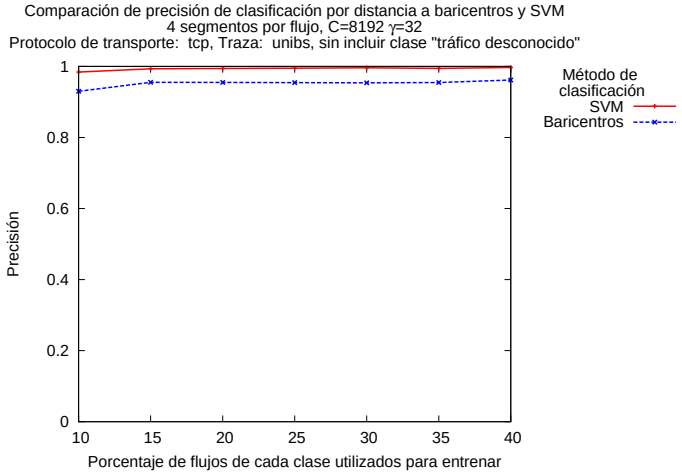

Figura 6.27

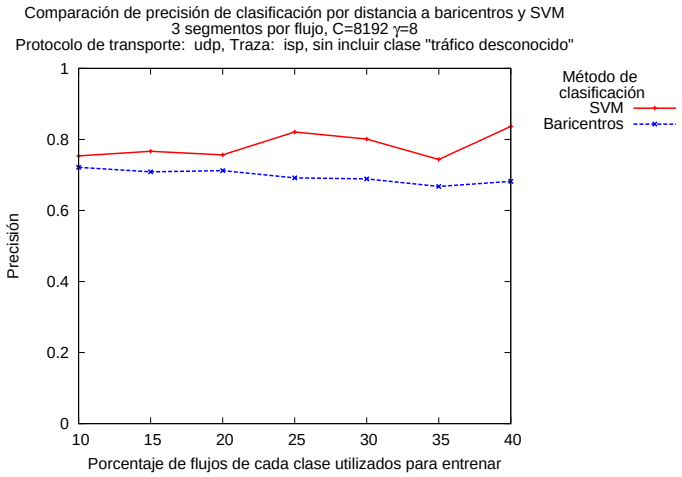

Figura 6.28

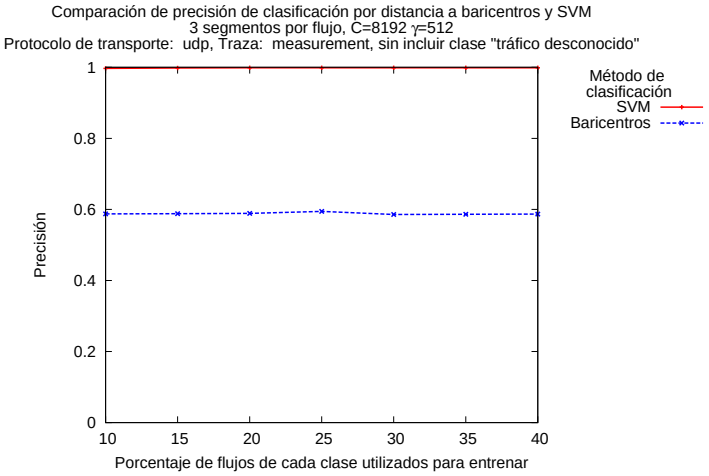

Figura 6.29

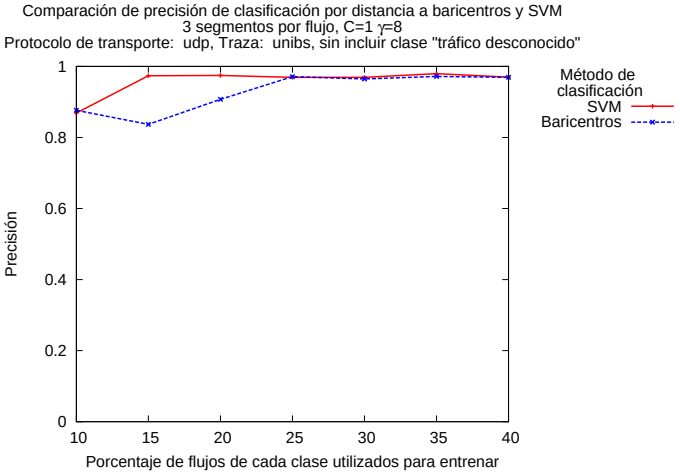

Figura 6.30

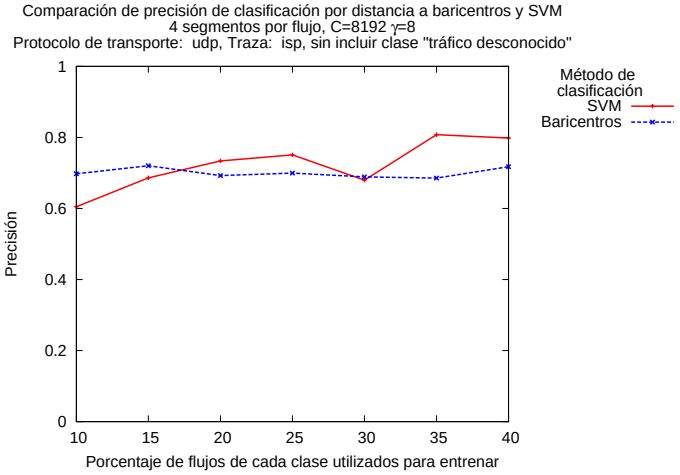

Figura 6.31

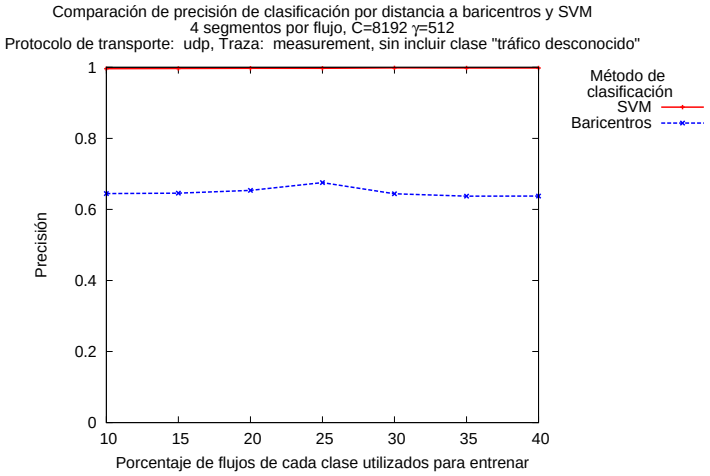

Figura 6.32

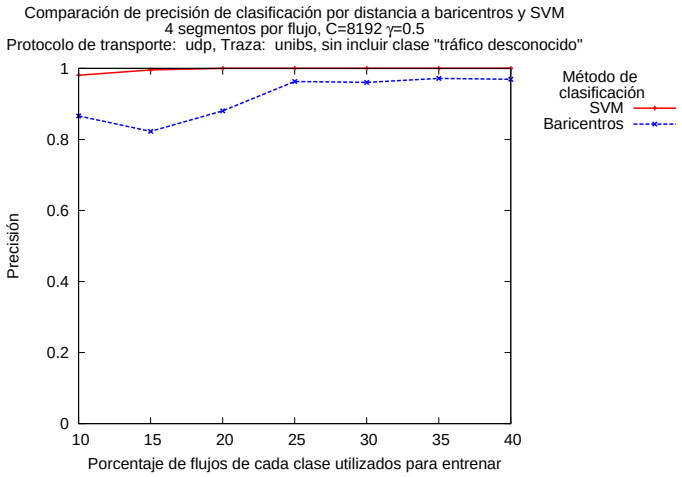
Comparación de precisión de clasificación por distancia a baricentros y SVM
4 segmentos por flujo, C=8192 γ=0.5
Protocolo de transporte: udp, Traza: unibs, sin incluir clase "tráfico desconocido"
Precisión
1
0.8
0.6
0.4
0.2
0
10
15
20
25
30
35
40
Porcentaje de flujos de cada clase utilizados para entrenar
Método de clasificación
SVM
Baricentros

Figura 6.33

6.3. Precisión por tipo de tráfico

6.3.1. Comparación de clasificación por distancia a baricentros y SVM

En esta sección se analiza el desempeño de los mecanismos de clasificación por distancia a baricentros y por SVM para cada tipo de tráfico o aplicación. Como se observa en las gráficas siguientes, aparecen diferencias importantes en la precisión de la clasificación entre las diferentes aplicaciones.

Estas diferencias se presentan tanto en la clasificación por distancia a baricentros como en la clasificación por SVM. Si bien la clasificación con SVM presenta niveles superiores de precisión que la clasificación por distancia a baricentros en la mayor parte de las aplicaciones, en algunos casos la precisión por clasificación a baricentros es superior.

Algunos de esos casos, por ejemplo la aplicación *ntp* en la traza *isp* 6.34, donde la clasificación por baricentros es mejor que la clasificación con SVM, seguramente sea atribuible a un error en la clasificación de base, ya que el protocolo estándar *ntp*[1] utiliza UDP como transporte.

En otros casos es interesante observar como la clasificación con SVM mejora de forma importante la clasificación por baricentros. Particularmente en el caso del protocolo *http* en la traza *measurement* puede verse la mejora obtenida con SVM así como el bajo nivel de la clasificación por distancia a baricentros, hecho que puede resultar intuitivo al observar la gráfica 6.2 donde se aprecia una importante dispersión de los flujos y por tanto puede preverse una baja precisión en la clasificación por baricentros.

Las gráficas permiten comparar la precisión por tipo de tráfico para ambos métodos. A efectos de no sobrecargar el documento, se muestran algunos ejemplos que abarcan las diferentes trazas, los diferentes protocolos de transporte (TCP y UDP) y considerando 3 o 4 segmentos para caracterizar los flujos.

En el Apéndice A se presentan los resultados completos –en el rango de valores analizado– de la clasificación para algunas aplicaciones seleccionadas, en forma de tabla, de modo que se pueda consultar algún caso que no aparece en las gráficas presentadas.

[1] *Network time protocol*

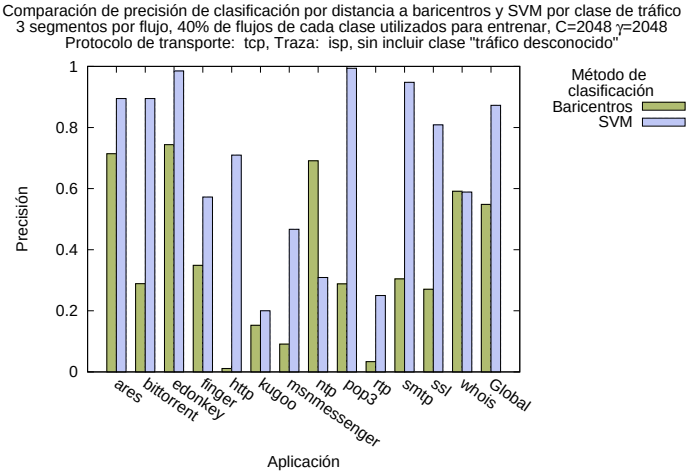

Figura 6.34

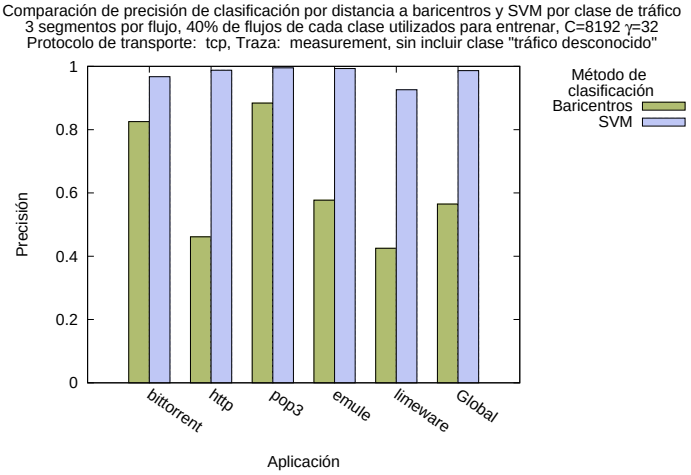

Figura 6.35

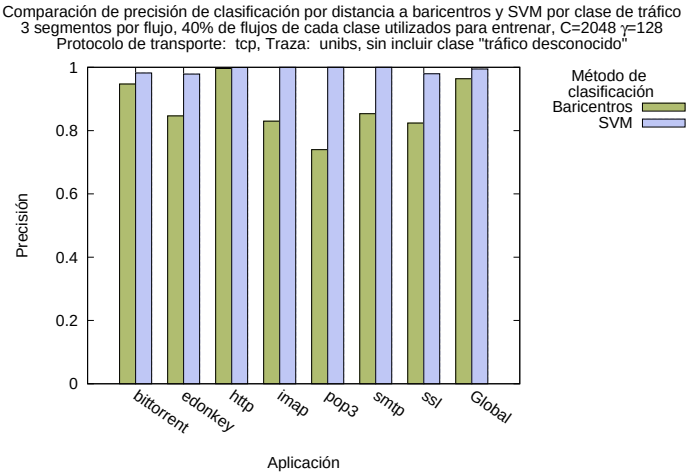

Figura 6.36

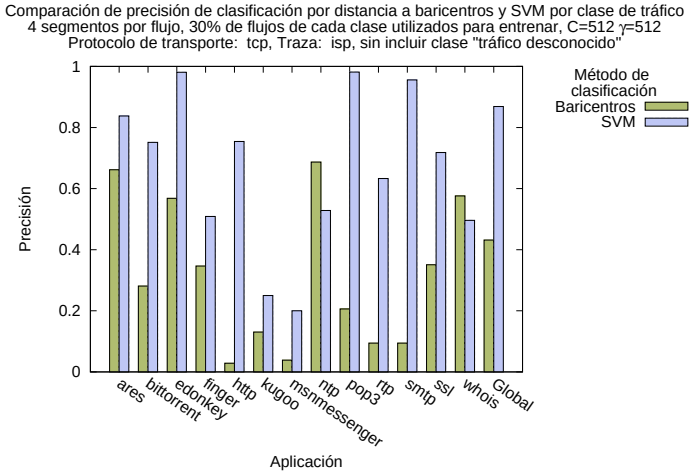

Figura 6.37

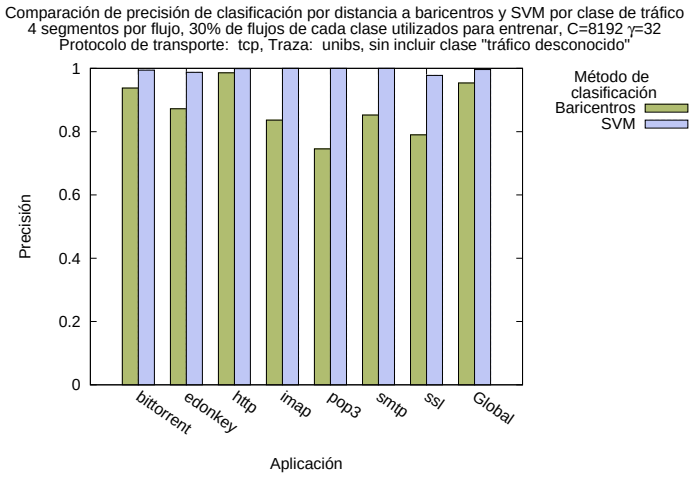

Figura 6.38

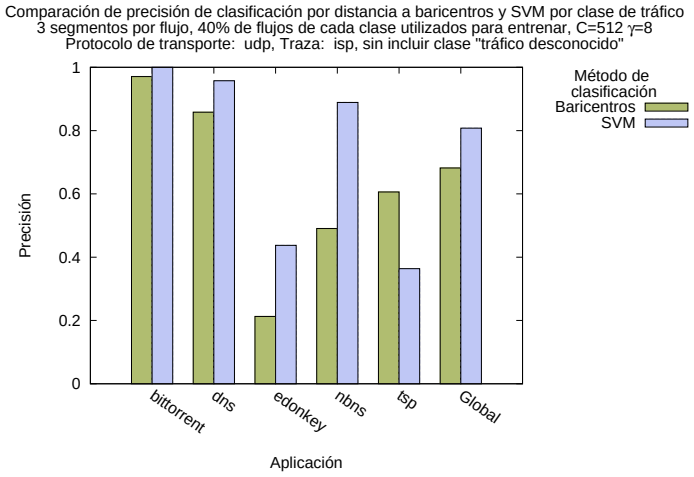

Figura 6.39

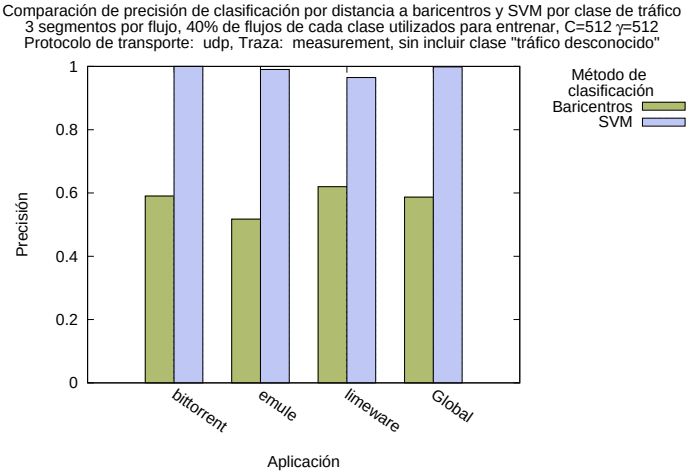

Figura 6.40

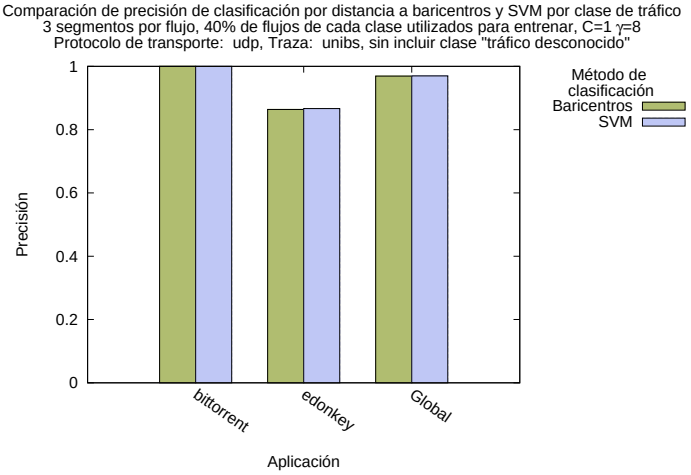

Figura 6.41

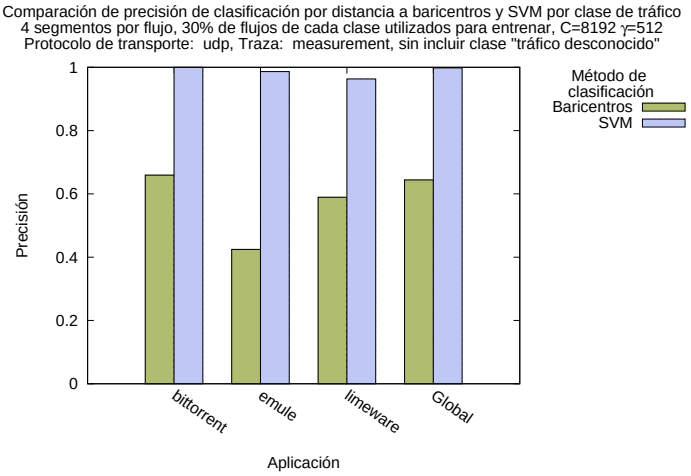

Figura 6.42

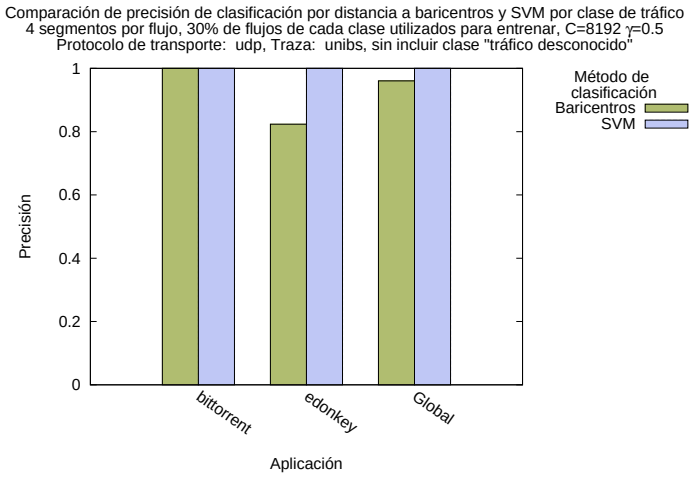
Comparación de precisión de clasificación por distancia a baricentros y SVM por clase de tráfico
4 segmentos por flujo, 30% de flujos de cada clase utilizados para entrenar, C=8192 γ=0.5
Protocolo de transporte: udp, Traza: unibs, sin incluir clase "tráfico desconocido"
Precisión
1
0.8
0.6
0.4
0.2
0
bittorrent
edonkey
Global
Aplicación
Método de clasificación
Baricentros
SVM

Figura 6.43

6.3.2. Votación de modelos SVM generados por *boosting*

En la clasificación por el método de *boosting* de modelos SVM, en primer lugar se generan modelos con el algoritmo *AdaBoost*, dando en cada paso mayor peso en la muestra de entrenamiento a los flujos que resultaron mal clasificados en el modelo del paso anterior. De este modo se pretende que los diferentes modelos SVM generados por este algoritmo se adapten mejor a diferentes tipo de tráfico previéndose que los flujos mal clasificados en un paso tengan mayor probabilidad de ser clasificados correctamente en el siguiente paso, ya que el modelo fue entrenado con más flujos de ese tipo.

En las siguientes gráficas, se muestran algunos ejemplos de la evolución de la precisión por tipo de aplicación en el proceso de *boosting* para la secuencia de modelos generados.

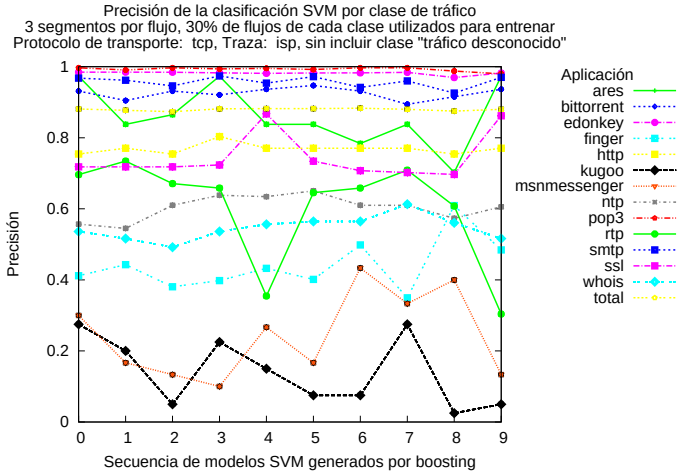

Figura 6.44

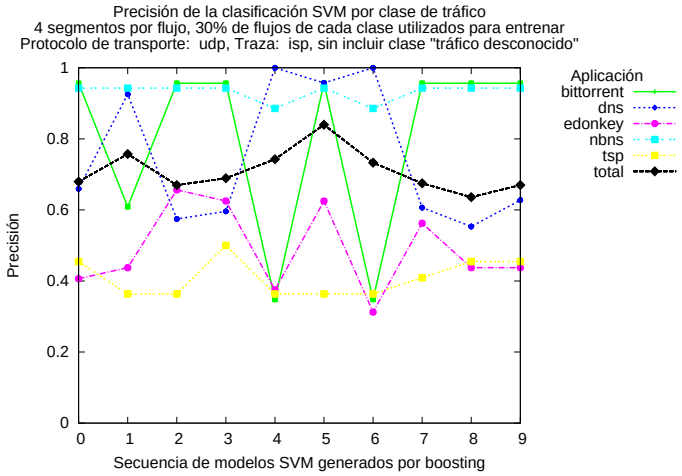

Figura 6.45

Se observa que los diferentes modelos SVM generados por el proceso de *boosting* obtienen precisiones diferentes para las diferentes aplicaciones,

manteniendo la precisión global de la muestra en niveles estables. Algunas aplicaciones mejoran notoriamente la precisión de su clasificación en algunos pasos de la secuencia.

La metodología de clasificación propuesta utilizando la técnica de *boosting* de SVM, consiste en realizar una votación ponderada del veredicto de cada modelo SVM de la secuencia. La ponderación propuesta es con el valor de confiabilidad de cada modelo para cada aplicación como se detalló en el capítulo 5.

6.3.3. Comparación de clasificación con SVM y votación por *boosting*

En esta sección se muestran algunas gráficas que comparan los resultados de la clasificación mediante SVM con la clasificación por el mecanismo de votación de modelos SVM. Se presentan para diferentes trazas de tráfico, utilizando 3 y 4 segmentos para representar los flujos. Se presentan resultados tanto para TCP como para UDP.

Se puede apreciar que la votación de modelos SVM generados por *boosting* permite mejorar la precisión de la clasificación para algunos tipos de tráfico. Asimismo en algunos casos empeora levemente, principalmente en aquellos que un solo modelo SVM logra altos valores de precisión. Se destaca asimismo que la precisión global se mantiene, ya que es un objetivo del algoritmo *AdaBoost*. Estas consideraciones permiten concluir que el mecanismo de *boosting* logra mejorar la precisión en algunos casos sin afectar mayormente los buenos resultados obtenidos por SVM.

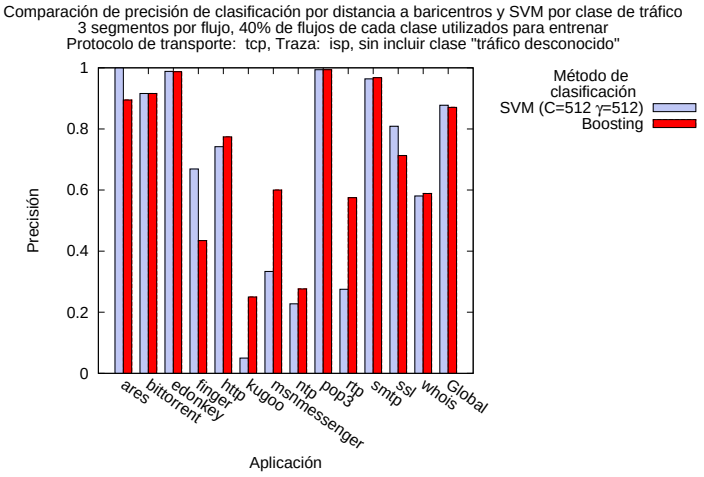

Figura 6.46

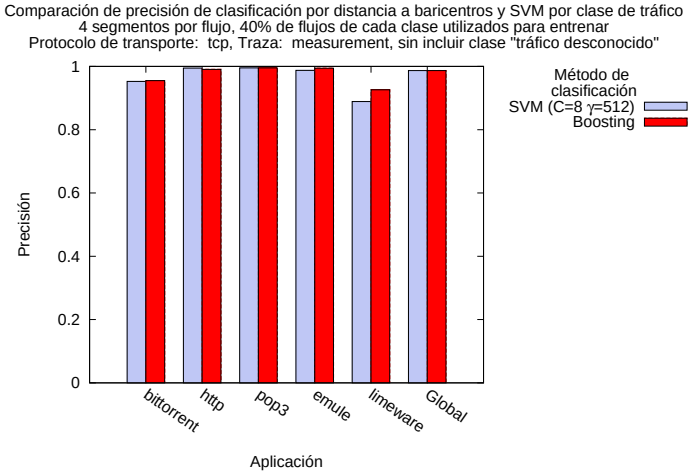

Figura 6.47

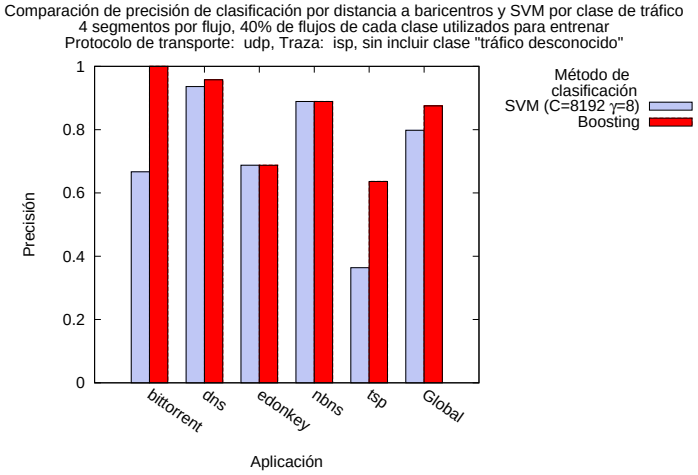

Figura 6.48

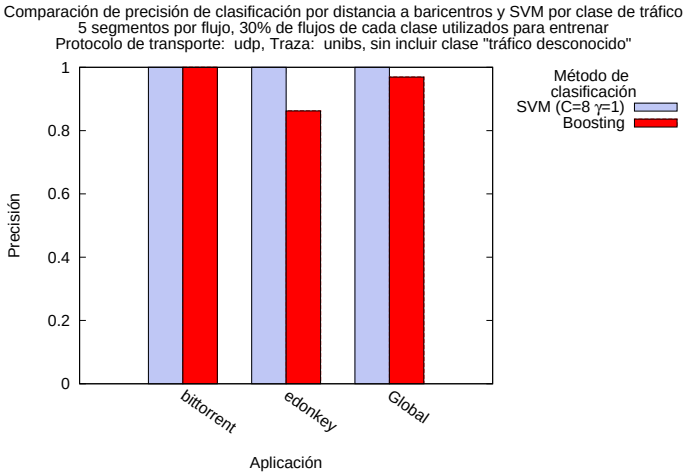

Figura 6.49

6.3.4. Comparación de clasificación por los tres métodos analizados

A continuación se muestran en forma conjunta los resultados de la precisión de la clasificación por tipo de tráfico para los tres métodos implementados. Se presentan los resultados tanto para TCP como UDP y para diferentes trazas de tráfico.

Los valores donde aún la precisión de la clasificación es superior utilizando el método de la distancia a baricentros se dan en la traza *isp* y podría deberse a las imprecisiones de la clasificación de base ya mencionada. En otras aplicaciones se obtienen mejoras en la precisión obtenida utilizando solamente un modelo SVM.

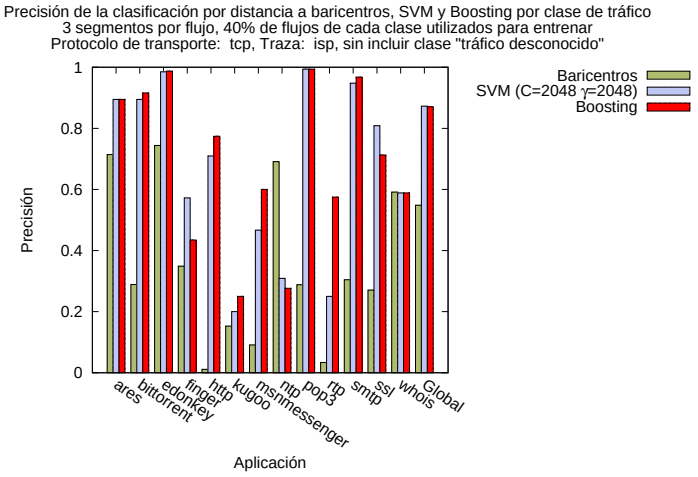

Figura 6.50

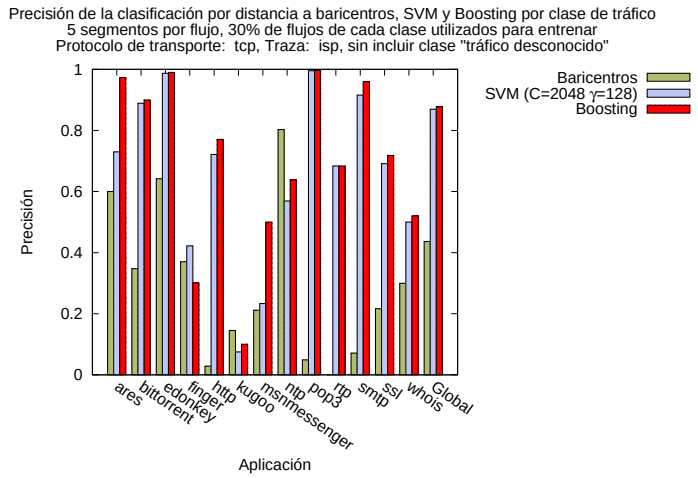

Figura 6.51

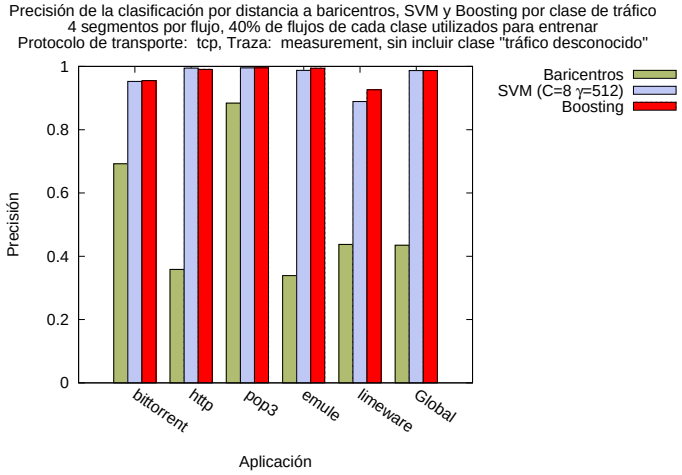

Figura 6.52

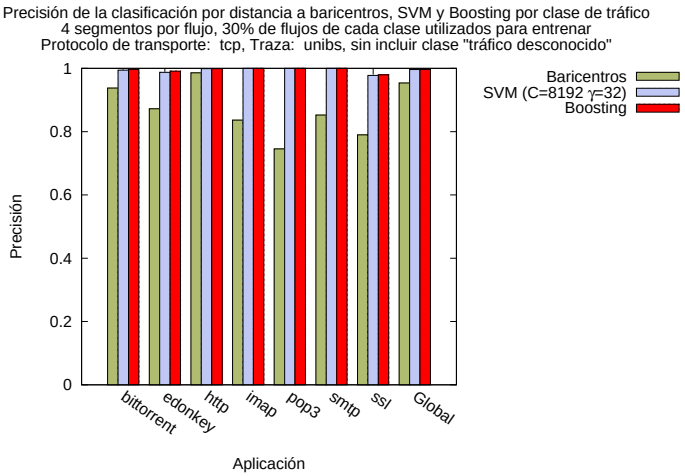

Figura 6.53

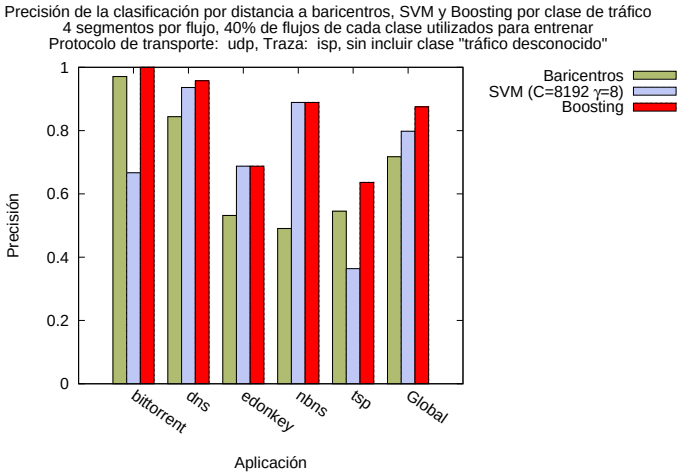

Figura 6.54

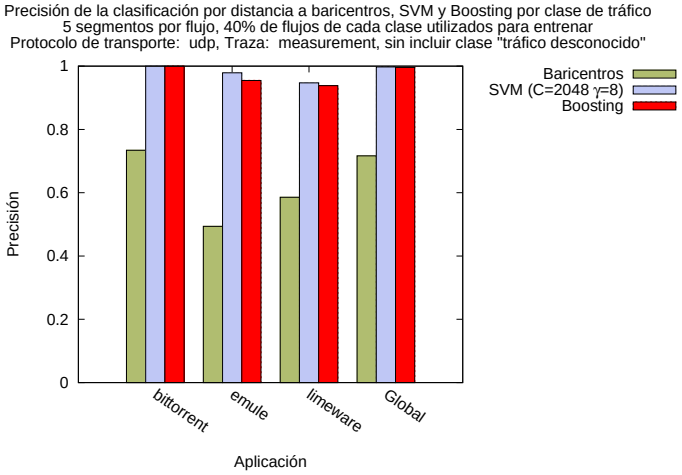

Figura 6.55

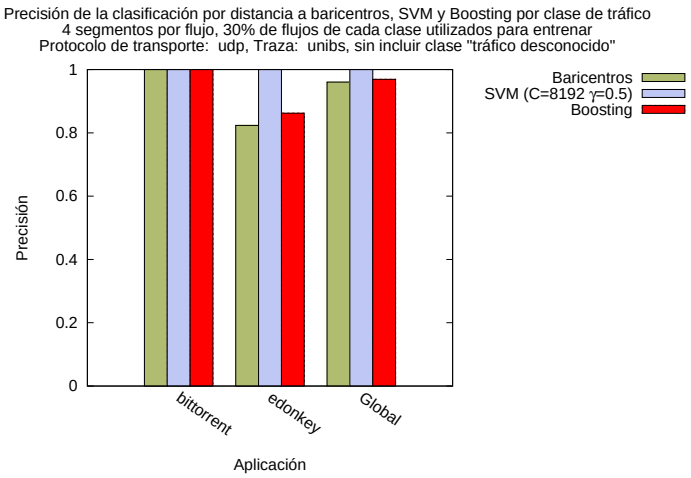

Figura 6.56

6.4. Efecto de la inclusión del tráfico desconocido

Como se mencionó en el capítulo 5 en este trabajo se propuso considerar el tráfico no identificado en las trazas disponibles como una clase más de tráfico. Las trazas *isp* y *unibs* contienen un porcentaje –en algunos casos importante– de tráfico desconocido, como se puede apreciar en las tablas 5.1, 5.2, 5.5, 5.6. Los resultados presentados en las secciones precedentes no incluyen estos flujos de tráfico desconocido, pero la metodología se aplicó tanto incluyendo como excluyendo dichos flujos, lo que permite realizar un análisis comparativo de la precisión.

En esta sección se comparan los resultados de precisión tanto para la clasificación por distancia a baricentros como para la clasificación con SVM, teniendo en cuenta la inclusión o no de la clase de tráfico desconocido. Solamente se presentan las gráficas para el caso de considerarse 4 segmentos por flujo y 40 % de flujos para entrenar.

Se puede observar algunas clases como *kugoo*, *ssl*, *edonkey*, *finger*, cambian notoriamente la precisión de su clasificación cuando se incluye el tráfico desconocido. Esas aplicaciones se confunden con el tráfico desconocido, al incluir este último como una clase más a clasificar. Esto permite pensar que la clasificación de base es incorrecta para estas aplicaciones. Esta observación es consistente con algunas observaciones que se realizaron cuando se desarrolló el modelo de clasificación con *L7-Filter*, donde se constató que algunos patrones de búsqueda (*finger*, *ssl*) debían considerarse como última opción luego de haber analizado los demás patrones posibles, ya que de lo contrario gran parte del tráfico era identificado como *finger* y *ssl*.

Del mismo modo, se puede observar que las restantes clases de tráfico mantienen prácticamente incambiados sus niveles de precisión con la inclusión del tráfico desconocido.

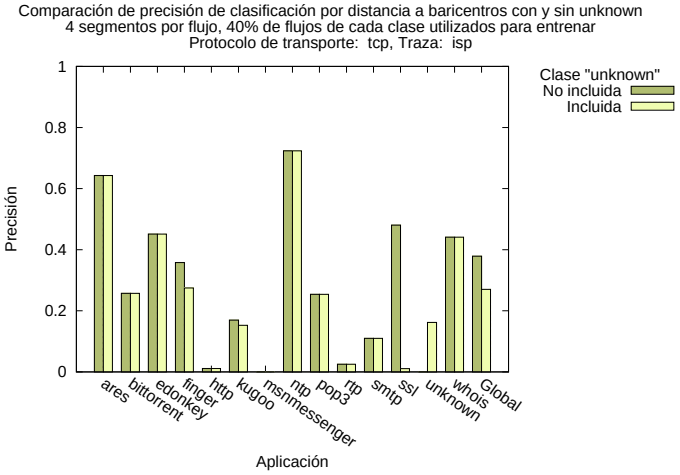

Figura 6.57

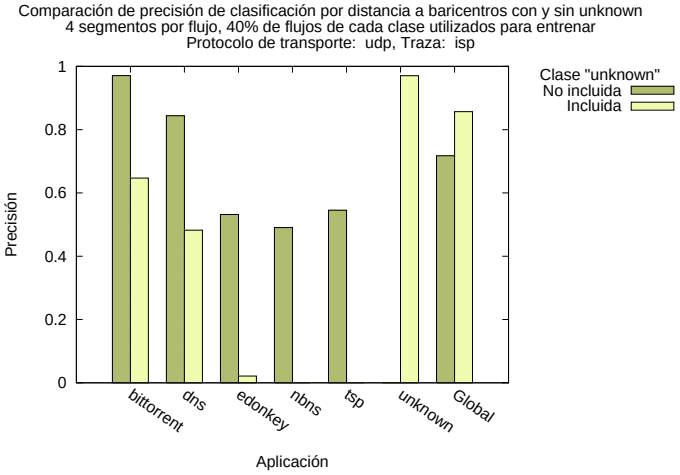

Figura 6.58

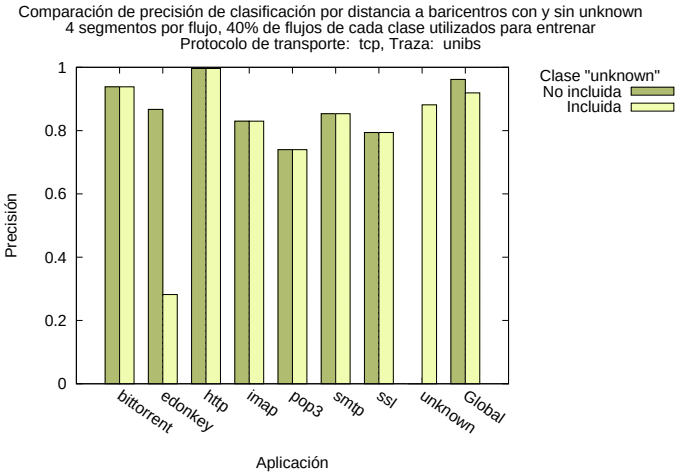

Figura 6.59

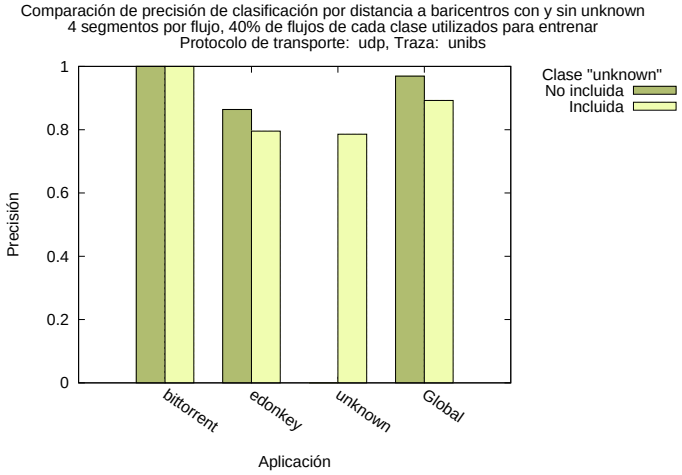

Figura 6.60

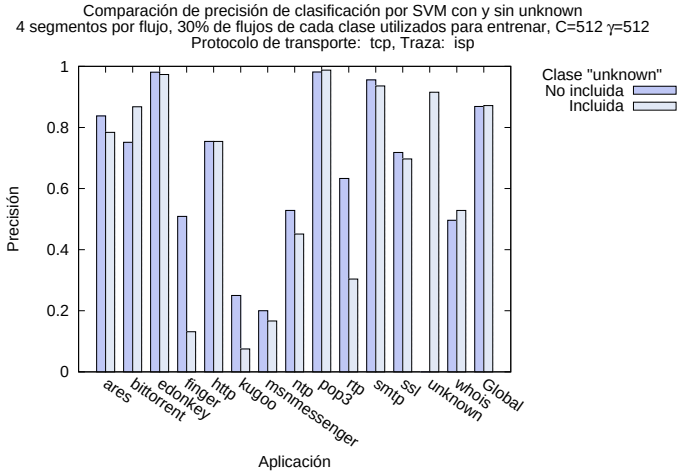

Figura 6.61

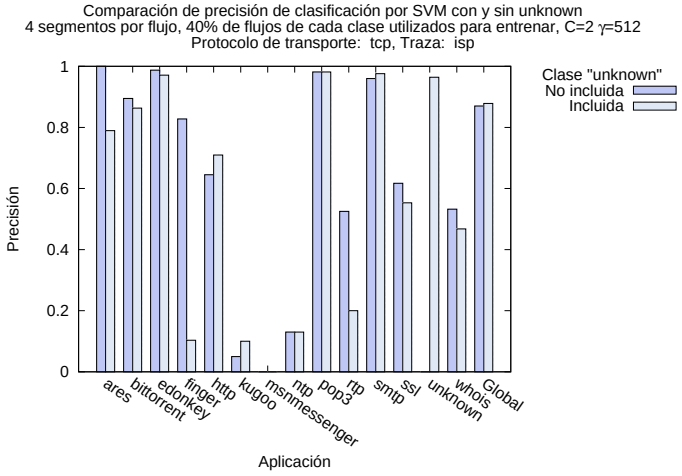

Figura 6.62

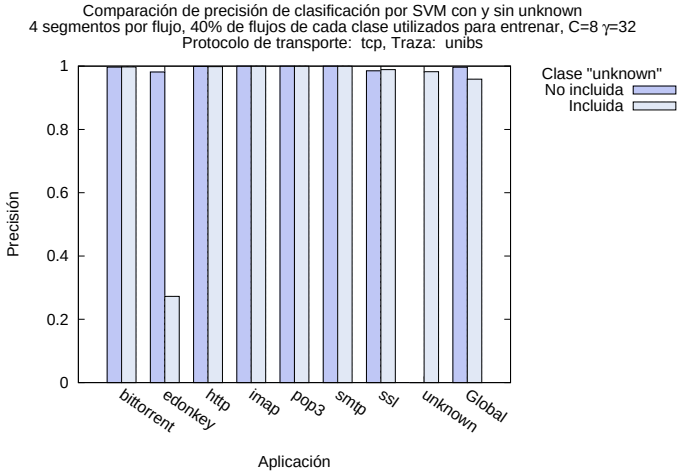

Figura 6.63

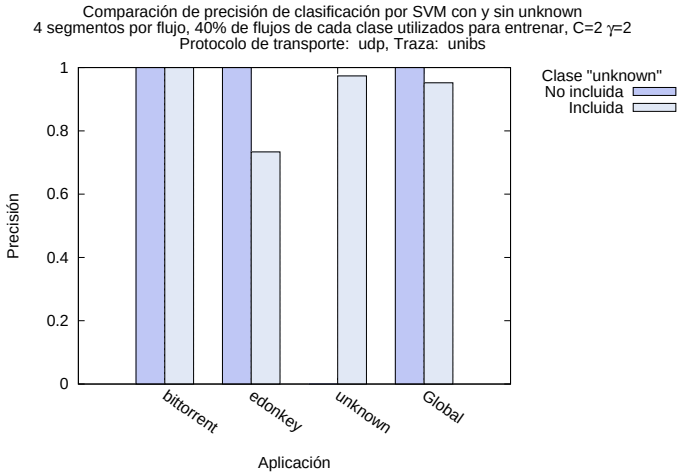

Figura 6.64

7 Conclusiones y trabajo futuro

7.1. Conclusiones

En este trabajo se abordó el problema de la clasificación de tráfico de Internet utilizando métodos estadísticos. En base al análisis de los resultados expuestos en este trabajo se pueden extraer algunas conclusiones.

La primera conclusión es que efectivamente es posible clasificar tráfico en base al análisis estadístico del tamaño de los primeros segmentos de los flujos de transporte sobre TCP o UDP. Los métodos analizados (distancia a baricentros y SVM) revelan en general una buena precisión en la clasificación utilizando esa información de los flujos, mostrando SVM mejores resultados.

Esta primer conclusión posibilita la clasificación de tráfico sin considerar la información de puertos bien conocidos y sin analizar datos que eventualmente sean sensibles para el usuario, ya que no es necesario tener acceso a los datos de capa de aplicación. Los tamaños de los segmentos, se obtienen a partir de campos disponibles en los encabezados de IP, TCP o UDP.

Asimismo, al considerarse solamente los primeros segmentos intercambiados, estas técnicas permitirían realizar una clasificación de tráfico en línea, ya que el tiempo de analizar solamente los primeros segmentos debería ser muy bajo si se dispone de un mecanismo de clasificación de rápida respuesta. Esto posibilitaría el uso de estas técnicas en conjunto con mecanismos de aplanamiento de tráfico, asignación de políticas de calidad de servicio o enrutamiento basado en políticas. Se han obtenido niveles altos de precisión global de la clasificación considerando solamente los primeros 3 o 4 segmentos de cada flujo.

En los resultados obtenidos, los valores de precisión global presentan una clara estabilidad cuando se utiliza entre 30 % y 40 % de los flujos para la fase de entrenamiento y notoriamente la técnica de SVM reporta una precisión global superior a la técnica de distancia a baricentros. Esta superioridad se observa tanto en el análisis de precisión global de una traza como en la precisión por tipo de aplicación para la mayor parte de las clases. Ajustando los

parámetros del modelo SVM se encuentran modelos que brindan precisiones globales superiores al 80 %.

La segunda conclusión importante es que la técnica de votación de modelos SVM generados mediante el algoritmo de *boosting*, propuesta para mejorar la precisión en algunas aplicaciones particulares sin perjudicar la precisión global, permite pensar en mecanismos de clasificación especialmente ajustados a cada tipo de tráfico que funcionen en forma colaborativa y que contribuyan en la decisión final a la hora de clasificar un flujo de tráfico desconocido.

Asimismo, los resultados obtenidos permiten concluir que los métodos de clasificación propuestos son válidos tanto para tráfico transportado sobre TCP como UDP. El comportamiento de la precisión global y por tipo de tráfico, así como la influencia de la cantidad de segmentos considerados y el tamaño de la muestra de entrenamiento, presenta resultados similares para ambos protocolos de transporte.

Finalmente, la influencia de entrenar las modelos SVM incluyendo o no el tráfico desconocido como una clase más, no refleja variaciones importantes en los resultados de precisión global y por tipo de aplicación. Particularmente en el análisis por tipo de tráfico, la clase "tráfico desconocido" o "*unknown*" presenta un comportamiento similar a las restantes clases. Sin embargo el análisis del tráfico desconocido permite presumir que algunas aplicaciones no están correctamente identificadas en la clasificación de base.

7.2. Trabajo Futuro

Un aspecto importante en relación con la aplicación de los mecanismos propuestos y que no ha sido abordado en este trabajo, es la performance de un eventual sistema de clasificación en línea por métodos estadísticos. Si bien los métodos propuestos serían aplicables a clasificación de tráfico en línea –por la información que utilizan para clasificar– es imprescindible analizar el desempeño de estos para que puedan considerarse como una solución implementable. La complejidad computacional relevante para la clasificación en línea es la correspondiente a la fase de verificación ya que la fase de entrenamiento se puede realizar fuera de línea y no presenta restricciones de tiempo real. Una posible línea de trabajo futuro sería la de analizar la performance de la fase de verificación del método SVM a efectos de determinar la capacidad de utilizarla para clasificación en línea. Dado que la fase de verificación con SVM consiste en la evaluación de una función, es de esperar que el costo no sea elevado, sin embargo se requiere un análisis específico para evaluarlo

correctamente. Del mismo modo, es necesario analizar el costo del proceso de *boosting* y de votación, en relación con la mejora de precisión que se obtenga.

Una objeción válida sobre el método propuesto, es que al clasificar en base a los primeros segmentos intercambiados por un flujo, el mecanismo podría ser evitado introduciendo algunas modificaciones a los protocolos de aplicación. Efectivamente si estos mecanismos comenzaran a desplegarse en la red, los diseñadores de los protocolos comenzarán a diseñar o re-diseñar los mismos de modo que se evite su detección o que resulten clasificados como una aplicación diferente. Sin embargo la historia de Internet está plagada de ejemplos similares. Las aplicaciones que hoy utilizan el puerto 80 pero que no corresponden al protocolo HTTP, surgieron a efectos de atravesar mecanismos de control como filtrado de paquetes en *firewalls* o mecanismos de *proxy* y esto llevó al diseño de nuevas herramientas de control que no dependan exclusivamente del análisis de los puertos de transporte. Otro ejemplo similar de un comportamiento por ciclos es el de los virus que afectan los equipos informáticos y los mecanismos de protección de virus (anti-virus) que se desarrollan por parte de la industria para detectarlos.

En este trabajo se analizó el comportamiento de los primeros segmentos intercambiados por los flujos de transporte considerando que se puede acceder a los segmentos en ambos sentidos del flujo. Esta restricción en principio puede parecer poco relevante, pero en las redes de los proveedores de servicio de Internet, es común que el camino de entrada y salida del tráfico no sea el mismo, debido a las características de multi-homing de los proveedores y las políticas de enrutamiento externo aplicables. Por tanto sería interesante analizar cómo podría clasificarse el tráfico contando solamente con una de las vías de los flujos. Este análisis podría realizarse en un trabajo futuro.

Finalmente, sería necesario profundizar el análisis de la aplicación del mecanismo de *boosting*, buscando algoritmos que permitan generar modelos ajustados a cada una de las aplicaciones. Del mismo modo, sería interesante analizar otras estrategias de votación o de ponderación de la votación.

Bibliografía

[1] G. Gómez Sena and P. Belzarena, "Early traffic classification using support vector machines," in *Proceedings of the 5th International Latin American Networking Conference*, ser. LANC '09. New York, NY, USA: ACM, 2009, pp. 60–66. [En línea] http://doi.acm.org/10.1145/1636682.1636693

[2] H.-C. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practices," in *ACM CoNEXT 2008*. [En línea] http://www.caida.org/publications/papers/2008/classification_demystified/

[3] A. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," in *Proceedings of the Passive y Active Measurement Workshop (PAM2005)*, March/Apri 2005, 2.

[4] S. Valenti, D. Rossi, M. Meo, M. Mellia, and P. Bermolen, "A behavioral classification framework for p2p-tv applications," TELECOM ParisTech (France), Politecnico di Torino (Italy), Tech. Rep. WP3.1, January 2009.

[5] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected means of protocol inference," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2006, pp. 313–326.

[6] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM, 2004, pp. 512–521.

[7] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 229–240, 2005.

[8] S. Fernandes, R. Antonello, T. Lacerda, A. Santos, D. Sadok, and T. Westholm, "Slimming down deep packet inspection systems," in

Proceedings of the 28th IEEE international conference on Computer Communications Workshops, ser. INFOCOM'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 61–66. [En línea] http://portal.acm.org/citation.cfm?id=1719850.1719861

[9] T. Hruby, K. van Reeuwijk, and H. Bos, "Ruler: easy packet matching and rewriting on network processors," in *Symposium on Architectures for Networking and Communications Systems (ANCS'07)*, Orlando, FL, USA, December 2007.

[10] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "Kiss: Stochastic packet inspection," in *Proceedings of the First International Workshop on Traffic Monitoring and Analysis*, ser. TMA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 117–125. [En línea] http://dx.doi.org/10.1007/978-3-642-01645-5_14

[11] A. Moore, M. Crogan, A. W. Moore, Q. Mary, D. Zuev, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," Tech. Rep., 2005.

[12] A. Dainotti, A. Pescape', and G. Ventre, "A packet-level characterization of network traffic," in *Proceedings of 2006 11th Intenational Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks*, 2006, pp. 38–45.

[13] R. D. . A. M. Wei Li, Kaysar Abdin, "Approaching real-time network traffic classification," Department of Computer Science, Queen Mary, University of London, Mile End Road, London E1 4NS, UK, Tech. Rep. RR-06-12, October 2006.

[14] R. Alshammari and A. Zincir-Heywood, "A preliminary performance comparison of two feature sets for encrypted traffic classification," in *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS 08*, ser. Advances in Soft Computing, E. Corchado, R. Zunino, P. Gastaldo, and l. Herrero, Eds. Springer Berlin / Heidelberg, 2009, vol. 53, pp. 203–210, 10.1007/978-3-540-88181-0_26. [En línea] http://dx.doi.org/10.1007/978-3-540-88181-0_26

[15] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 5–16, 2007.

[16] W. Li, M. Canini, A. W. Moore, and R. Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Comput. Netw.*, vol. 53, pp. 790–809, April 2009. [En línea] http://portal.acm.org/citation.cfm?id=1517860.1518209

[17] M. Ichino, H. Maeda, T. Yamashita, K. Hoshi, N. Komatsu, K. Takeshita, M. Tsujino, M. Iwashita, and H. Yoshino, "Internet traffic classification using score level fusion of multiple classifier," in *Proceedings of the 2010 IEEE/ACIS 9th International Conference on Computer and Information Science*, ser. ICIS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 105–110. [En línea] http://dx.doi.org/10.1109/ICIS.2010.48

[18] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques." in *PAM '04*, 2004, pp. 205–214.

[19] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, June 2005. [En línea] http://dx.doi.org/10.1145/1071690.1064220

[20] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for internet traffic classification," *IEEE Transaction on Neural Networks*, vol. 18, pp. 223–239, 2007.

[21] A. Dainotti, W. D. Donato, A. Pescapè, and P. S. Rossi, "Classification of network traffic via packet-level hidden markov models," in *Global Telecommunications Conference*, 2008, pp. 2138–2142.

[22] J. Erman, A. Mahanti, and M. Arlitt, "Offline/online traffic classification using semi-supervised learning," Perform. Eval, Tech. Rep., 2007.

[23] Y. Zeng and T. M. Chen, "Classification of traffic flows into qos classes by unsupervised learning and knn clustering," *TIIS*, vol. 3, no. 2, pp. 134–146, 2009.

[24] W. Li and A. W. Moore, "Learning for accurate classification of real-time traffic," in *Proceedings of the 2006 ACM CoNEXT conference*, ser. CoNEXT '06. New York, NY, USA: ACM, 2006, pp. 36:1–36:2. [En línea] http://doi.acm.org/10.1145/1368436.1368480

[25] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Perform. Eval.*, vol. 67, pp. 451–467, June 2010. [En línea] http://dx.doi.org/10.1016/j.peva.2010.01.001

[26] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, April 2006. [En línea] http://dx.doi.org/10.1145/1129582.1129589

[27] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *CoNEXT '06: Proceedings of the 2006 ACM CoNEXT conference*. New York, NY, USA: ACM, 2006, pp. 1–12.

[28] N.-F. Huang, G.-Y. Jai, and H.-C. Chao, "Early identifying application traffic with application characteristics," May 2008, pp. 5788–5792.

[29] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for qos: A statistical signature-based approach to ip traffic classification," in *In IMC 04*, 2004, pp. 135–148.

[30] "Net neutrality as global principle for internet governance." [En línea] http://www.internetgovernance.org/pdf/NetNeutralityGlobalPrinciple.pdf

[31] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008. [En línea] http://dx.doi.org/10.1109/SURV.2008.080406

[32] Y. dar Lin, C. nan Lu, Y. cheng Lai, W. hao Peng, and P. ching Lin, "Application classification using packet size distribution and port association," *Journal of Network and Computer Applications*, vol. 32, pp. 1023–1030, 2009.

[33] G. Maiolini, G. Molina, A. Baiocchi, and A. Rizzi, "On the fly application flows identification by exploiting k-means based classifiers," 2009. [En línea] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.150.2767

[34] Z. Li, R. Yuan, and X. Guan, "Accurate classification of the internet traffic based on the svm method," in *ICC*, 2007, pp. 1373–1378.

[35] Y. Liu, H. Liu, H. Zhang, and X. Luan, "The internet traffic classification an online svm approach," in *Information Networking, 2008. ICOIN 2008. International Conference on*, 2008, pp. 1 –5.

[36] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for tcp traffic classification," *Computer Networks and Isdn Systems*, vol. 53, pp. 2476–2490, 2009.

[37] A. El Gonnouni, A. Lyhyaoui, J. Antari, and S. El Jelali, "Support vector machine for internet traffic identification," in *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*, dec. 2007, pp. 351 –354.

[38] F. Liu, Z. Li, and Q. Nie, "A new method of p2p traffic identification based on support vector machine at the host level," in *Proceedings of the 2009 International Conference on Information Technology and Computer Science - Volume 02*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 579–582. [En línea] http://portal.acm.org/citation.cfm?id=1588295.1588595

[39] G. Lu, H. Zhang, X. Sha, C. Chen, and L. Peng, "Tcfom: A robust traffic classification framework based on oc-svm combined with mc-svm," in *Communications and Intelligence Information Security (ICCIIS), 2010 International Conference on*, oct. 2010, pp. 180 –186.

[40] Y. xiang Yang, R. Wang, Y. Liu, S. zhen Li, and X. yong Zhou, "Solving p2p traffic identification problems via optimized support vector machines," in *AICCSA*, 2007, pp. 165–171.

[41] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti, "Abacus: Accurate behavioral classification of p2p-tv traffic," *Computer Networks*, vol. 55, no. 6, pp. 1394 – 1411, 2011. [En línea] http://www.sciencedirect.com/science/article/pii/S1389128610003713

[42] R. Wang, Y. Liu, Y. xiang Yang, and H. long Wang, "A new method for p2p traffic identification based on support vector machine," in *Proc. of the second ICGST International Conference on Artificial Intelligence and Machine Learning AIML 05*, A. A. et al., Ed., vol. 06, ICGST. Sharm El-Sheikh, Egypt: ICGST, June 2006, pp. 58–63.

[43] A. Marnerides, D. Pezaros, H. Kim, and D. Hutchison, "Unsupervised two-class and multi-class support vector machines for abnormal traffic characterization," in *10th International Passive and Active Measurements Conference (PAM 2009)*, April 2009. [En línea] http://eprints.gla.ac.uk/34546/

[44] L. Cheng, J. Zhang, J. Yang, and J. Ma, "An improved adaboost algorithm based on adaptive weight adjusting," in *Proceedings of the 3rd international conference on Advanced Data Mining and Applications*, ser. ADMA '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 625–632. [En línea] http://dx.doi.org/10.1007/978-3-540-73871-8_60

[45] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern Recognition*, vol. 36, no. 12, pp. 2757 – 2767, 2003. [En línea] http://www.sciencedirect.com/science/article/B6V14-49202D3-3/2/fca47360665ecf4dd30f1b4b0f01151e

[46] X. Zhang and F. Ren, "Improving svm learning accuracy with adaboost," in *Proceedings of the 2008 Fourth International Conference on Natural Computation - Volume 03*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 221–225. [En línea] http://portal.acm.org/citation.cfm?id=1473245.1473612

[47] X. Zan, J. Han, J. Zhang, Q. Zheng, and C. Han, "A boosting approach for intrusion detection," *Journal of Electronics (China)*, vol. 24, pp. 369–373, 2007, 10.1007/s11767-005-0201-z. [En línea] http://dx.doi.org/10.1007/s11767-005-0201-z

[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] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[50] H. W. Kuhn and 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.

[51] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[52] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, August 1997. [En línea] http://portal.acm.org/citation.cfm?id=261540.261549

[53] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[54] "The perl programming language." [En línea] http://www.perl.org/

[55] "Unibs: Data sharing, available traces with associated ground truth." [En línea] http://www.ing.unibs.it/ntw/tools/traces/index.php

[56] G. Szabó, D. Orincsay, S. Malomsoky, and I. Szabó, "On the validation of traffic classification algorithms," ser. PAM'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 72–81. [En línea] http://portal.acm.org/citation.cfm?id=1791949.1791960

[57] "tcpdump, a powerful command-line packet analyzer; and libpcap, a portable c/c++ library for network traffic capture." [En línea] http://www.tcpdump.org/

[58] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and k. c. claffy, "Gt: picking up the truth from the ground for internet traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 12–18, October 2009. [En línea] http://doi.acm.org/10.1145/1629607.1629610

[59] J. Zhang and A. Moore, "Traffic trace artifacts due to monitoring via port mirroring," in *E2EMON*, K. Saraç and T. Friedman, Eds. IEEE, 2007, pp. 1–8.

[60] M. Canini, W. Li, A. W. Moore, and R. Bolla, "Gtvs: Boosting the collection of application traffic ground truth," in *Proceedings of the First International Workshop on Traffic Monitoring and Analysis*, ser. TMA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 54–63. [En línea] http://dx.doi.org/10.1007/978-3-642-01645-5_7

[61] A. Dainotti, W. Donato, and A. Pescapé, "Tie: A community-oriented traffic classification platform," in *Proceedings of the First International Workshop on Traffic Monitoring and Analysis*, ser. TMA '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 64–74. [En línea] http://dx.doi.org/10.1007/978-3-642-01645-5_8

[62] D. Saí, S. Hauger, and M. Köhn, "Architecture and scalability of a high-speed traffic measurement platform with a highly flexible packet classification," *Comput. Netw.*, vol. 53, pp. 810–820, April 2009. [En línea] http://portal.acm.org/citation.cfm?id=1517860.1518210

[63] "Application layer packet classifier for linux (l7-filter)." [En línea] http://17-filter.sourceforge.net/

[64] "Userspace command line program used to configure the linux 2.4.x and 2.6.x ipv4 packet filtering ruleset." [En línea] http://www.netfilter.org/projects/iptables/index.html

[65] "Bit-twist is a simple and powerful libpcap-based ethernet packet generator." [En línea] http://bittwist.sourceforge.net/

[66] C. W. Hsu, C. C. Chang, and 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

Índice de tablas

Apéndices

Apéndice A

Tablas de resultados de precisión global

En esta sección se presentan las tablas completas de resultados de precisión global obtenida por los métodos de distancia a baricentros y clasificación con SVM, a efectos de visualizar la ganancia del método de SVM. Se muestran los valores resultantes para las tres trazas analizadas.

Los resultados abarcan un rango de 2 a 6 segmentos por flujo y de 10 a 40 % de variación de flujos de entrenamiento. En el caso de SVM se indica el valor de los parámetros C y γ con los que se obtiene el valor de precisión indicado.

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Mejora de SVM sobre baricentros (%)
			C	γ	Precisión (%)	
2	10	18.19	8192	512	85.79	371.51
2	15	12.49	8192	2048	87.40	599.74
2	20	12.63	2048	2048	88.13	597.57
2	25	12.90	2048	8192	88.09	582.81
2	30	52.82	8192	2048	88.39	67.33
2	35	55.28	2048	8192	87.55	58.38
2	40	55.71	8192	8192	86.52	55.29
3	10	55.68	512	512	86.91	56.10
3	15	55.20	512	2048	87.37	58.27
3	20	54.20	8192	2048	87.27	61.02
3	25	53.95	512	512	87.87	62.87
3	30	52.77	2048	512	88.09	66.95
3	35	54.86	8192	2048	87.29	59.13
3	40	54.84	8192	2048	87.14	58.92
4	10	53.94	512	512	86.59	60.55
4	15	53.12	2048	128	86.64	63.10
4	20	49.69	2048	512	86.81	74.70
4	25	50.50	512	512	87.86	73.99
4	30	43.14	512	512	86.86	101.33
4	35	40.42	2048	512	87.63	116.80
4	40	37.89	2048	512	86.32	127.83
5	10	50.41	512	32	86.96	72.51
5	15	51.21	8192	128	87.66	71.18
5	20	48.52	8192	32	86.34	77.93
5	25	48.09	2048	128	87.62	82.21
5	30	43.63	2048	128	86.93	99.23
5	35	39.70	512	512	87.00	119.18
5	40	39.01	8192	128	87.02	123.06
6	10	48.60	512	32	85.55	76.05
6	15	49.08	8192	32	86.70	76.66
6	20	47.27	2048	128	86.26	82.50
6	25	46.75	8192	128	87.06	86.23
6	30	41.38	8192	32	87.05	110.37
6	35	38.09	8192	128	86.58	127.32
6	40	37.00	128	128	85.81	131.93

Tabla A.1: Precisión de la clasificación global para tráfico TCP en la traza *isp*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Mejora de SVM sobre baricentros (%)
			C	γ	Precisión (%)	
2	10	71.49	8	8	71.46	-0.04
2	15	76.61	8	32	75.83	-1.01
2	20	75.61	8192	32	84.74	12.08
2	25	67.62	8192	32	86.38	27.75
2	30	75.56	8	8192	82.04	8.58
2	35	67.07	8192	8	82.05	22.34
2	40	70.13	2048	32	84.62	20.66
3	10	72.14	8	8	72.68	0.76
3	15	70.87	8	8	76.67	8.18
3	20	71.22	8	512	80.19	12.60
3	25	69.17	2048	1	71.60	3.50
3	30	68.89	8192	32	81.55	18.38
3	35	66.77	8	8192	78.85	18.09
3	40	68.18	8	2048	82.69	21.28
4	10	69.76	8192	0.5	73.90	5.93
4	15	72.02	8192	2	74.44	3.37
4	20	69.27	8	8192	75.32	8.74
4	25	69.95	8192	8	75.10	7.36
4	30	68.89	8192	8	67.96	-1.35
4	35	68.56	8192	8	80.77	17.80
4	40	71.75	8192	8	79.81	11.23
5	10	61.77	8192	0.5	67.32	8.98
5	15	71.56	8	2	74.72	4.42
5	20	70.98	8192	2	72.08	1.55
5	25	70.73	8192	8	84.44	19.39
5	30	71.39	8192	8	67.48	-5.48
5	35	66.47	8192	8	79.49	19.59
5	40	69.16	8192	8	85.58	23.75
6	10	64.36	8	1	80.24	24.67
6	15	71.56	8	8	73.06	2.09
6	20	72.68	8	2	75.65	4.08
6	25	72.02	512	0.5	81.71	13.46
6	30	73.89	8192	2	64.56	-12.62
6	35	67.66	2048	2	82.69	22.21
6	40	71.43	8	2	86.54	21.15

Tabla A.2: Precisión de la clasificación global para tráfico UDP en la traza *isp*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Mejora de SVM sobre baricentros (%)
			C	γ	Precisión (%)	
2	10	90.93	8192	2048	92.11	1.29
2	15	90.58	8192	2048	98.65	8.90
2	20	90.20	8192	8192	98.97	9.73
2	25	89.81	8192	8192	99.23	10.49
2	30	89.90	8192	8192	99.24	10.40
2	35	90.42	8192	8192	99.39	9.91
2	40	91.17	8192	8192	99.44	9.07
3	10	95.62	2048	128	98.76	3.28
3	15	95.68	8192	128	98.92	3.38
3	20	95.84	8192	128	99.41	3.72
3	25	95.74	8192	128	99.29	3.70
3	30	96.58	8192	128	99.31	2.82
3	35	96.34	2048	512	99.49	3.26
3	40	96.38	8192	512	99.54	3.27
4	10	93.00	2048	32	98.39	5.80
4	15	95.51	128	32	99.30	3.97
4	20	95.50	2048	32	99.43	4.12
4	25	95.44	512	32	99.57	4.33
4	30	95.38	8192	32	99.61	4.43
4	35	95.46	2048	128	99.49	4.22
4	40	96.16	8192	32	99.68	3.66
5	10	93.04	8192	2	98.22	5.56
5	15	93.13	8192	8	99.12	6.43
5	20	95.52	2048	8	99.27	3.92
5	25	95.41	2048	8	99.28	4.06
5	30	95.56	512	32	99.50	4.13
5	35	95.71	2048	32	99.43	3.89
5	40	95.84	8192	32	99.53	3.85
6	10	92.76	2048	8	97.71	5.33
6	15	92.74	8192	8	98.72	6.45
6	20	92.64	8192	8	98.80	6.64
6	25	95.06	512	8	99.18	4.33
6	30	95.17	2048	8	99.30	4.34
6	35	95.33	8192	8	99.23	4.10
6	40	95.54	8192	8	99.34	3.98

Tabla A.3: Precisión de la clasificación global para tráfico TCP en la traza *unibs*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Mejora de SVM sobre baricentros (%)
			C	γ	Precisión (%)	
2	10	90.03	8	8	99.23	10.21
2	15	87.68	8	8	99.12	13.04
2	20	96.53	8	8	98.97	2.54
2	25	97.12	8	8	98.77	1.69
2	30	96.48	8	8	98.46	2.06
2	35	97.16	8	8	97.96	0.83
2	40	96.92	8	8	96.97	0.05
3	10	87.63	8	2	97.30	11.03
3	15	83.70	8	8	97.36	16.32
3	20	90.73	8	8	97.44	7.39
3	25	97.12	8	8	96.91	-0.21
3	30	96.48	8	8	96.92	0.46
3	35	97.16	8	8	97.96	0.83
3	40	96.92	8	8	96.97	0.05
4	10	86.60	8	2	97.30	12.36
4	15	82.25	8	8	97.36	18.37
4	20	88.03	8	8	97.95	11.27
4	25	96.30	8	1	100.00	3.85
4	30	96.04	8192	0.5	100.00	4.13
4	35	97.16	8	8	97.96	0.83
4	40	96.92	8192	0.5	100.00	3.17
5	10	83.51	8	8	95.75	14.67
5	15	80.43	8	8	97.36	21.04
5	20	86.87	8192	0.5	100.00	15.11
5	25	96.30	8	8	96.30	0.00
5	30	96.04	8	1	100.00	4.13
5	35	95.73	8	8	96.94	1.26
5	40	94.87	8	8	96.97	2.21
6	10	83.51	8	2	96.14	15.13
6	15	80.43	8	8	97.80	21.59
6	20	84.17	8	8	96.92	15.15
6	25	92.59	8	8	96.91	4.67
6	30	94.27	8	8	96.92	2.81
6	35	94.79	8192	0.5	100.00	5.50
6	40	93.85	8	8	93.94	0.10

Tabla A.4: Precisión de la clasificación global para tráfico UDP en la traza *unibs*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Mejora de SVM sobre baricentros (%)
			C	γ	Precisión (%)	
2	10	72.69	8192	512	97.75	34.48
2	15	74.35	8192	2048	98.20	32.08
2	20	62.08	8192	2048	98.22	58.22
2	25	62.16	8192	512	98.49	58.45
2	30	62.27	8192	2048	98.58	58.31
2	35	62.07	512	8192	98.96	59.44
2	40	62.24	8192	8192	98.86	58.83
3	10	74.01	512	32	97.65	31.95
3	15	64.65	512	128	98.20	51.90
3	20	54.05	128	32	98.22	81.72
3	25	54.28	8	128	98.54	81.55
3	30	54.34	8192	128	98.70	81.64
3	35	56.34	128	128	98.74	75.25
3	40	56.47	2048	32	98.83	75.01
4	10	66.54	128	32	97.74	46.90
4	15	47.14	512	32	97.92	107.71
4	20	47.27	2048	128	97.95	107.22
4	25	47.49	128	32	98.67	107.79
4	30	41.42	8	512	98.50	137.80
4	35	49.12	8192	128	98.74	101.03
4	40	43.48	8	512	98.67	126.94
5	10	56.12	8192	32	96.89	72.64
5	15	40.99	8192	8	97.47	137.80
5	20	40.87	2048	32	97.42	138.37
5	25	41.38	128	32	98.23	137.38
5	30	43.39	8192	128	98.18	126.24
5	35	44.58	8192	128	98.39	120.72
5	40	44.68	2048	128	98.60	120.68
6	10	50.54	8192	8	96.91	91.76
6	15	40.47	8192	32	97.29	140.42
6	20	40.30	32	32	97.21	141.23
6	25	41.38	2048	32	97.81	136.38
6	30	42.78	8	128	97.93	128.92
6	35	42.92	2048	32	98.31	129.02
6	40	42.89	2048	128	98.15	128.83

Tabla A.5: Precisión de la clasificación global para tráfico TCP en la traza *measurement*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Mejora de SVM sobre baricentros (%)
			C	γ	Precisión (%)	
2	10	71.95	8192	128	99.87	38.80
2	15	71.93	8	512	99.89	38.87
2	20	71.89	8	512	99.89	38.96
2	25	71.99	8192	128	99.90	38.77
2	30	71.92	8192	512	99.89	38.89
2	35	72.01	8192	128	99.89	38.72
2	40	72.02	8192	128	99.87	38.68
3	10	58.75	8192	8	99.71	69.72
3	15	58.80	8192	8	99.83	69.78
3	20	58.90	8192	512	99.85	69.54
3	25	59.46	8192	128	99.88	67.99
3	30	58.58	8192	128	99.90	70.53
3	35	58.63	8192	512	99.86	70.32
3	40	58.69	8192	512	99.85	70.12
4	10	64.47	8192	128	99.75	54.73
4	15	64.56	8192	32	99.74	54.50
4	20	65.37	8	128	99.78	52.65
4	25	67.53	128	128	99.82	47.81
4	30	64.42	8192	512	99.82	54.96
4	35	63.73	8192	128	99.85	56.68
4	40	63.77	8192	8	99.76	56.43
5	10	70.52	8192	32	99.68	41.35
5	15	71.26	8	32	99.71	39.92
5	20	71.75	512	32	99.78	39.06
5	25	73.19	8192	32	99.81	36.37
5	30	71.90	128	128	99.78	38.79
5	35	71.87	2048	32	99.79	38.84
5	40	71.65	2048	8	99.74	39.21
6	10	67.00	2048	8	99.69	48.79
6	15	67.37	512	32	99.70	47.98
6	20	70.37	512	32	99.66	41.63
6	25	71.98	8192	32	99.78	38.61
6	30	68.41	2048	8	99.73	45.79
6	35	68.16	512	32	99.79	46.41
6	40	67.78	8192	32	99.81	47.25

Tabla A.6: Precisión de la clasificación global para tráfico UDP en la traza *measurement*

Tablas de resultados de precisión por tipo de aplicación

En esta sección se presentan, a modo de ejemplo, los resultados para algunas aplicaciones, comparándose los métodos de clasificación por baricentros, con un modelo SVM y por votación de modelos SVM generados por *boosting*.

Las tablas permiten analizar para qué valores se logran mejoras con el mecanismo de *boosting* y cómo no se afectan mayormente los buenos valores de clasificación con un modelo SVM.

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Precisión boosting	Mejora de SVM sobre baricentros (%)	Mejora de boosting sobre SVM
			C	γ	Precisión (%)			
2	10	2.94	8192	512	68.59	70.25	2232.23	2.41
2	15	0.78	8192	2048	75.47	72.64	9635.85	-3.75
2	20	0.83	2048	2048	74.73	78.02	8941.76	4.41
2	25	0.88	2048	8192	80.26	81.58	9050.00	1.64
2	30	3.77	8192	2048	80.33	78.69	2028.69	-2.04
2	35	3.03	2048	8192	71.74	76.09	2267.39	6.06
2	40	3.30	8192	8192	74.19	77.42	2150.54	4.35
3	10	82.35	512	512	65.29	73.55	-20.72	12.66
3	15	2.33	512	2048	70.75	75.47	2942.45	6.67
3	20	0.83	8192	2048	70.33	78.02	8409.89	10.94
3	25	1.75	512	512	81.58	82.89	4550.00	1.61
3	30	0.94	2048	512	75.41	80.33	7893.44	6.52
3	35	2.02	8192	2048	73.91	78.26	3558.69	5.88
3	40	1.10	8192	2048	70.97	77.42	6358.06	9.09
4	10	78.68	512	512	66.12	66.12	-15.97	0.00
4	15	4.65	2048	128	69.81	72.64	1400.94	4.05
4	20	19.83	2048	512	74.73	75.82	276.74	1.47
4	25	2.63	512	512	77.63	81.58	2850.00	5.08
4	30	2.83	512	512	75.41	78.69	2564.48	4.35
4	35	2.02	2048	512	73.91	76.09	3558.69	2.94
4	40	1.10	2048	512	64.52	74.19	5770.97	15.00
5	10	76.47	512	32	67.77	66.94	-11.38	-1.22
5	15	5.43	8192	128	67.92	72.64	1151.75	6.94
5	20	21.49	8192	32	73.63	75.82	242.65	2.99
5	25	3.51	2048	128	77.63	81.58	2112.50	5.08
5	30	2.83	2048	128	72.13	77.05	2448.63	6.82
5	35	1.01	512	512	65.22	73.91	6356.52	13.33
5	40	1.10	8192	128	64.52	74.19	5770.97	15.00
6	10	24.26	512	32	66.12	68.60	172.48	3.75
6	15	2.33	8192	32	72.64	75.47	3023.58	3.90
6	20	1.65	2048	128	72.53	75.82	4287.91	4.55
6	25	1.75	8192	128	78.95	78.95	4400.00	-0.00
6	30	2.83	8192	32	73.77	73.77	2506.56	-0.00
6	35	3.03	8192	128	65.22	69.57	2052.17	6.67
6	40	3.30	128	128	67.74	67.74	1954.84	0.00

Tabla A.7: Precisión de la clasificación de *http* (TCP) en la traza *isp*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Precisión boosting	Mejora de SVM sobre baricentros (%)	Mejora de boosting sobre SVM
			C	γ	Precisión (%)			
2	10	90.61	8192	512	99.88	99.88	10.24	-0.00
2	15	90.71	8192	2048	99.87	99.87	10.09	-0.00
2	20	90.71	8192	2048	99.85	99.85	10.07	0.00
2	25	90.46	8192	512	99.81	99.81	10.34	0.00
2	30	90.72	8192	2048	99.77	99.77	9.98	-0.00
2	35	90.57	512	8192	99.69	99.69	10.07	0.00
2	40	91.02	8192	8192	99.54	99.54	9.36	0.00
3	10	88.24	512	32	99.88	99.88	13.20	-0.00
3	15	88.31	512	128	99.87	99.87	13.09	-0.00
3	20	88.27	128	32	99.85	99.85	13.11	0.00
3	25	87.86	8	128	99.81	99.81	13.61	0.00
3	30	88.20	8192	128	99.77	99.77	13.12	-0.00
3	35	88.00	128	128	99.69	99.69	13.28	0.00
3	40	88.39	2048	32	99.54	99.54	12.61	0.00
4	10	88.24	128	32	99.88	99.88	13.20	-0.00
4	15	88.31	512	32	99.87	99.87	13.09	-0.00
4	20	88.27	2048	128	99.85	99.85	13.11	0.00
4	25	87.86	128	32	99.81	99.81	13.61	0.00
4	30	88.20	8	512	99.77	99.77	13.12	-0.00
4	35	88.00	8192	128	100.00	99.69	13.64	-0.31
4	40	88.39	8	512	99.54	99.54	12.61	0.00
5	10	88.24	8192	32	99.88	99.88	13.20	-0.00
5	15	88.31	8192	8	99.87	99.87	13.09	-0.00
5	20	88.27	2048	32	99.85	99.85	13.11	0.00
5	25	87.86	128	32	99.81	99.81	13.61	0.00
5	30	88.20	8192	128	99.77	99.77	13.12	-0.00
5	35	88.00	8192	128	99.69	99.69	13.28	0.00
5	40	88.39	2048	128	99.54	99.54	12.61	0.00
6	10	84.93	8192	8	99.88	99.88	17.60	-0.00
6	15	84.92	8192	32	99.87	99.87	17.60	-0.00
6	20	85.02	32	32	99.85	99.85	17.44	0.00
6	25	84.76	2048	32	99.81	99.81	17.76	0.00
6	30	85.15	8	128	99.77	99.77	17.17	-0.00
6	35	84.86	2048	32	99.69	99.69	17.48	0.00
6	40	85.29	2048	128	99.54	99.54	16.70	0.00

Tabla A.8: Precisión de la clasificación de *pop3* (TCP) en la traza *measurement*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Precisión boosting	Mejora de SVM sobre baricentros (%)	Mejora de boosting sobre SVM
			C	γ	Precisión (%)			
2	10	55.38	8	8	96.55	96.55	74.33	0.00
2	15	45.16	8	8	96.08	96.08	112.75	0.00
2	20	84.48	8	8	95.45	95.45	12.99	0.00
2	25	87.04	8	8	94.44	94.44	8.51	0.00
2	30	84.31	8	8	93.10	93.10	10.42	0.00
2	35	87.23	8	8	90.91	90.91	4.21	-0.00
2	40	86.36	8	8	86.67	86.67	0.35	-0.00
3	10	44.62	8	2	87.93	87.93	97.09	0.00
3	15	27.42	8	8	88.24	92.16	221.80	4.44
3	20	58.62	8	8	88.64	90.91	51.20	2.56
3	25	87.04	8	8	86.11	88.89	-1.06	3.23
3	30	84.31	8	8	86.21	86.21	2.25	-0.00
3	35	87.23	8	8	90.91	90.91	4.21	-0.00
3	40	86.36	8	8	86.67	86.67	0.35	-0.00
4	10	40.00	8	2	87.93	87.93	119.83	0.00
4	15	20.97	8	8	88.24	90.20	320.81	2.22
4	20	46.55	8	8	90.91	90.91	95.29	-0.00
4	25	83.33	8	1	100.00	88.89	20.00	-11.11
4	30	82.35	8192	0.5	100.00	86.21	21.43	-13.79
4	35	87.23	8	8	90.91	90.91	4.21	-0.00
4	40	86.36	8192	0.5	100.00	86.67	15.79	-13.33
5	10	26.15	8	8	81.03	84.48	209.84	4.26
5	15	12.90	8	8	88.24	90.20	583.82	2.22
5	20	41.38	8192	0.5	100.00	90.91	141.67	-9.09
5	25	83.33	8	8	83.33	88.89	-0.00	6.67
5	30	82.35	8	1	100.00	86.21	21.43	-13.79
5	35	80.85	8	8	86.36	90.91	6.82	5.26
5	40	77.27	8	8	86.67	86.67	12.16	-0.00
6	10	26.15	8	2	82.76	74.14	216.43	-10.42
6	15	12.90	8	8	90.20	90.20	599.02	-0.00
6	20	29.31	8	8	86.36	90.91	194.65	5.26
6	25	66.67	8	8	86.11	88.89	29.17	3.23
6	30	74.51	8	8	86.21	86.21	15.70	-0.00
6	35	76.60	8192	0.5	100.00	90.91	30.56	-9.09
6	40	72.73	8	8	73.33	86.67	0.83	18.18

Tabla A.9: Precisión de la clasificación de *edonkey* (UDP) en la traza *unibs*

Segmentos por flujo	% de flujos de entrenamiento	Precisión por dist. a baricentros (%)	Clasificación con SVM			Precisión boosting	Mejora de SVM sobre baricentros (%)	Mejora de boosting sobre SVM
			C	γ	Precisión (%)			
2	10	86.32	8	8	77.13	76.06	-10.65	-1.38
2	15	84.50	8	32	73.33	76.97	-13.22	4.96
2	20	82.98	8192	32	95.04	98.58	14.53	3.73
2	25	83.05	8192	32	93.22	95.76	12.24	2.73
2	30	81.82	8	8192	100.00	97.87	22.22	-2.13
2	35	80.39	8192	8	95.77	98.59	19.13	2.94
2	40	85.11	2048	32	100.00	95.74	17.50	-4.26
3	10	93.87	8	8	73.40	75.53	-21.80	2.90
3	15	96.00	8	8	77.58	73.33	-19.19	-5.47
3	20	94.15	8	512	97.87	73.76	3.95	-24.64
3	25	92.09	2048	1	72.88	72.88	-20.86	-0.00
3	30	90.30	8192	32	96.81	92.55	7.20	-4.40
3	35	83.66	8	8192	100.00	98.59	19.53	-1.41
3	40	85.82	8	2048	100.00	95.74	16.53	-4.26
4	10	91.51	8192	0.5	70.74	78.72	-22.69	11.28
4	15	95.50	8192	2	66.67	75.15	-30.19	12.73
4	20	88.83	8	8192	100.00	72.34	12.57	-27.66
4	25	87.01	8192	8	69.49	78.81	-20.13	13.41
4	30	86.06	8192	8	65.96	72.34	-23.36	9.68
4	35	79.74	8192	8	91.55	91.55	14.81	-0.00
4	40	84.40	8192	8	93.62	95.74	10.92	2.27
5	10	93.40	8192	0.5	63.30	92.02	-32.23	45.38
5	15	96.00	8	2	90.91	73.33	-5.30	-19.33
5	20	94.15	8192	2	68.09	71.63	-27.68	5.21
5	25	93.22	8192	8	95.76	69.49	2.73	-27.43
5	30	92.73	8192	8	54.26	86.17	-41.49	58.82
5	35	78.43	8192	8	88.73	91.55	13.13	3.17
5	40	82.27	8192	8	97.87	97.87	18.97	0.00
6	10	97.17	8	1	96.28	78.72	-0.92	-18.23
6	15	95.00	8	8	79.39	76.97	-16.43	-3.05
6	20	94.15	8	2	74.47	73.05	-20.90	-1.90
6	25	92.66	512	0.5	88.14	73.73	-4.88	-16.35
6	30	93.33	8192	2	58.51	60.64	-37.31	3.64
6	35	78.43	2048	2	94.37	92.96	20.32	-1.49
6	40	81.56	8	2	97.87	100.00	20.00	2.17

Tabla A.10: Precisión de la clasificación de *dns* (UDP) en la traza *isp*

Esta es la última página de la tesis.
Versión del 16 de diciembre de 2011.