Acceso a servicios basado en modelado de Markov: eDonkey como caso de estudio

Rafael A. Rodríguez-Gómez
Dpto. de Teoría de la Señal, Telemática y Comunicaciones
CITIC, Universidad de Granada
Email: rodgom@ugr.es

Gabriel Maciá-Fernández
Dpto. de Teoría de la Señal, Telemática y Comunicaciones
CITIC, Universidad de Granada
Email: gmacia@ugr.es

Pedro García-Teodoro
Dpto. de Teoría de la Señal, Telemática y Comunicaciones
CITIC, Universidad de Granada
Email: pgteodor@ugr.es

***Resumen*—La gestión y monitorización de tráfico es un tema de especial relevancia para los operadores de red. Tanto a los proveedores de servicio (ISPs) como a los administradores de red les interesa controlar el uso de sus recursos en función del servicio accedido por los usuarios. En este trabajo, se presenta una aproximación estocástica basada en modelos de Markov para detectar servicios específicos y, a partir de ello, posibilitar el control de acceso a los recursos de la red. El sistema presentado trabaja a nivel de flujo considerando los paquetes como observaciones entrantes y es capaz de analizar tanto comunicaciones encriptadas como no encriptadas. En primera instancia se presenta una estructura general para modelar cualquier servicio de red, el cual es posteriormente aplicado al protocolo eDonkey como caso de estudio.**

Para validar nuestra aproximación se evalúa el sistema con trazas de tráfico real, evidenciando los resultados experimentales la bondad de nuestra aproximación para alcanzar los objetivos pretendidos.

I. Introducción

La monitorización y clasificación de tráfico supone una herramienta altamente útil para los proveedores de servicio y administradores de sistemas con objeto de controlar los accesos a los recursos de la red. Y ello, tanto con fines de planificación y mantenimiento como de cara a la provisión de seguridad en los servicios y sistemas [1] [2].

Tres son los aspectos principales a resaltar de un proceso típico de clasificación de tráfico:

- *Parametrización de tráfico*: Multitud de características han sido utilizadas en la literatura para representar el tráfico de red, desde la salida de los enrutadores SNMP relativa a las estadísticas de sesión [3] hasta las cabeceras TCP incluyendo los *bits* de señalización y los primeros *bytes* de contenido [4].
- *Nivel de identificación*: Una vez el tráfico ha sido parametrizado, se consideran en la literatura principalmente dos niveles diferentes para realizar la clasificación o identificación [5]: identificación basada en flujos e identificación basada en paquetes. En la identificación basada en flujos el objetivo principal es clasificar cada flujo como perteneciente o no a un servicio dado, mientras que la identificación basada en paquetes pretende clasificar cada paquete de forma individual.
- *Proceso de identificación*: Finalmente, los esquemas utilizados para llevar a cabo la identificación cubren un amplio rango de técnicas. Desde simples heurísticas o indicadores [6] hasta complejas técnicas de minería de datos o reconocimiento de patrones [7].

En relación a los aspectos presentados con anterioridad, este artículo presenta una aproximación eficiente para detectar determinado tráfico y con ello el acceso a determinados servicios. Las características más destacables de nuestra aproximación son:

1) En primer lugar, en el nivel de identificación se considera una aproximación basada en flujos. Para esto, se utilizan los puertos origen y destino de cada paquete a fin de definir un flujo o comunicación. Posteriormente, el flujo completo se modela e identifica como perteneciente o no a un servicio dado.
2) Para realizar la clasificación a nivel de flujo, cada paquete se considera una observación del sistema. Con este fin, cada paquete se parametriza como un vector tridimensional: $<psize, itime, chdir>$, donde *psize* es el tamaño del contenido del paquete, *itime* el tiempo entre llegadas con respecto a la recepción del paquete anterior del flujo, y *chdir* es una variable que indica el cambio de dirección del paquete (IP1 $\rightarrow$ IP2 o IP2 $\rightarrow$ IP1) con respecto al paquete previo en el flujo.
3) Finalmente, la clasificación de un flujo se basa en la consideración un modelo de Markov que representa las comunicaciones pertenecientes a un servicio dado. Así, será posible diferenciar entre tráfico HTTP, DNS o BitTorrent, entre otros.

El resto del artículo se organiza como sigue. La Sección II presenta los trabajos relacionados especialmente en el campo de la clasificación de tráfico mediante el uso de modelos de Markov. Posteriormente, la parametrización y estructura del modelo de Markov específicamente utilizados en nuestro caso para la clasificación de tráfico se presentan en la Sección IV. La evaluación de esta aproximación para el control de acceso se lleva a cabo en la Sección V. Finalmente, las principales conclusiones extraídas de este trabajo se muestran en la Sección VI.

II. Trabajos relacionados

La mayoría de la investigación existente en el campo de la clasificación de tráfico puede dividirse en tres grupos: *(i)*

basada en puertos conocidos, *(ii)* basada en el contenido de los paquetes y *(iii)* basada en las características de los flujos. Algunos estudios ponen de manifiesto la escasa efectividad de la identificación basada en puertos para el tráfico actual [3]. Posteriormente las técnicas más extendidas fueron las basadas en el contenido de los paquetes [8]. Dadas las limitaciones de las técnicas anteriores la comunidad investigadora ha centrado sus esfuerzos en el desarrollo de técnicas basadas en las características de los flujos [9].

En el presente trabajo, se propone una técnica para controlar el acceso a ciertos servicios de una red por medio de una identificación de tráfico basada en características de los flujos y en el uso de modelos de Markov. Son pocas las contribuciones existentes en la bibliografía similares a la aquí propuesta, de manera que seguidamente se presentan las más significativas y se comparan con la específica aquí introducida.

Los autores en [10] utilizan modelos de Markov para representar el comportamiento de un flujo específico, siendo las observaciones del modelo los paquetes de control, *p.ej.* SYN, ACK, SYN-ACK, PSH-ACK, PSH, etc. Estas observaciones son radicalmente diferentes a las escogidas en el presente trabajo. Por otra parte, para obtener buenos resultados de detección esta aproximación requiere de un número elevado de paquetes de control. Incluso con un número alto los resultados obtenidos presentan una elevada tasa de falsos positivos; por ejemplo, con 50 paquetes de control en un flujo existe un 10% de falsos positivos al intentar diferenciar HTTP de HTTPS. En cambio, nuestra aproximación es independiente del número de paquetes en el flujo, obteniendo mejores resultados de detección.

Wright et al. [11] siguen un diseño similar al utilizado en el alineamiento de proteínas. Los autores utilizan un modelo oculto de markov (HMM) de izquierda a derecha con un número de estados igual al número medio de paquetes con contenido en los flujos del servicio a detectar. La principal diferencia con nuestra aproximación es la topología seleccionada para el HMM. Los resultados obtenidos son muy variables dependiendo del servicio objetivo; de 58,20% a 92,90% para la tasa de aciertos y de 7,90% a 0,62% para los falsos positivos. Como se mostrará en la sección de resultados experimentales, nuestra aproximación supera estos resultados.

En [12] los autores utilizan HMMs para clasificar protocolos basados en TCP en un estado temprano, utilizando para ello entre los 4 y 10 primeros paquetes de cada flujo. Alcanzan tasas de acierto relativamente bajas (en torno al 70%) y sólo pueden detectar protocolos sustentados sobre TCP.

El modelo propuesto en [13] por Dainotti et al. presenta un modelo diferente para cada servicio a detectar. Un punto débil de esta propuesta es el reducido tamaño de las bases de datos de tráfico utilizadas. Por ejemplo, para el tráfico del juego Age of Mitology se utilizan 4 flujos para entrenar y 2 para evaluar. Nosotros utilizamos más de 240.000 flujos eDonkey, alcanzando mejores tasas de acierto y similares de falsos positivos.

III. Fundamentos del modelado de Markov

En teoría de probabilidad, el modelado de Markov se refiere a un modelo que asume la propiedad de Markov para un proceso dado. Esto quiere decir que la probabilidad de estados futuros depende únicamente del estado actual.

El modelado de Markov ha sido aplicado a multitud de campos, desde reconocimiento del habla a medicina, en sismología y en ingeniería. Comúnmente, los dos tipos principales de modelos utilizados son: cadenas de Markov y HMM. En el primer caso, cada estado del sistema es modelado a través de una variable aleatoria que cambia en función del tiempo. Un HMM, por su parte, es una cadena de Markov en la que el estado es parcialmente observable. Las observaciones están relacionadas con el estado pero no son información suficiente como para determinar con precisión el estado al que pertenecen. Los HMM son la técnica predominante en clasificación [14] y es en éstos en los que nos centraremos en el resto de la sección.

A. Conceptos generales en HMM

Dada una cadena de Markov discreta con un conjunto de estados finito $S = \{s_1, s_2, ..., s_N\}$, se define un HMM como la tupla $\lambda = (\Pi, A, B)$, donde

1) $\Pi = \{\pi_1, \pi_2, ..., \pi_N\}$ es el vector de probabilidades iniciales, esto es la probabilidad de que el estado *i* sea el primero en la secuencia: $\pi_i = P(q_1 = s_i), i \in [1, ..., N]$
2) $A = [a_{ij}]$ es la matriz de probabilidades de transición. Cada término representa la probabilidad de transición del estado *i* en el instante t al *j* en el instante $t + 1$: $a_{ij} = P(q_{t+1} = s_j | q_t = s_i), i, j \in [1, ..., N]$
3) $B = [b_{jk}]$ es la matriz de probabilidades de observación, esto es la probabilidad de que la observación k se produzca en el estado i en el instante t: $b_{ik} = P(o_t = v_k | q_t = s_i), i \in [1, ..., N], k \in [1, ..., M]$

B. Clasificación basada en HMM

La clasificación con HMM implica resolver dos cuestiones: la *decodificación* y el *entrenamiento*. Decodificar se refiere a encontrar la secuencia de estados Q de λ asociada a la secuencia O observada. Para esto se utiliza el algoritmo de Viterbi:

$$Q = \underset{Q'}{argmax} P(Q'|O, \lambda) = \underset{Q'}{argmax} P(O|Q', \lambda)$$

donde

$$P(O|Q', \lambda) = \pi_{q_1} b_{q_1 o_1} \prod_{t=2}^{T} a_{q_{t-1} q_t} b_{q_t o_t} \quad (1)$$

Por otro lado, el entrenamiento pretende encontrar los parámetros del modelo: Π, A y B. El algoritmo más utilizado para ello es Baum-Welch.

Para reducir el tamaño del espacio de observación y, por tanto, la complejidad de la parametrización de las observaciones, se utiliza la cuantización vectorial (VQ). Dos aspectos

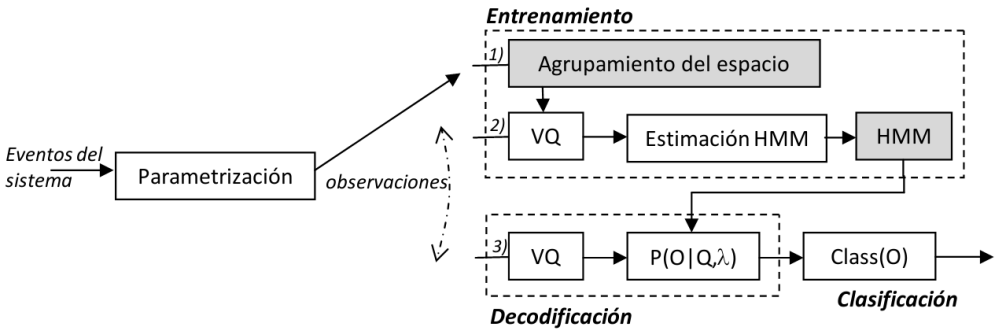

Fig. 1: Esquema general para la clasificación basada en HMM

a tener en cuenta en la cuantización vectorial son: el algoritmo de cuantización utilizado (mayoritariamente K-medias) y la medida de distancia para determinar la proximidad de dos observaciones (la Euclídea es la más extendida).

Resumiendo, una clasificación basada en HMM sigue el esquema general presentado en la Fig. 1:

1) Cada evento del sistema se parametriza utilizando un vector multidimensional.
2) Inicialmente se realiza una etapa de entrenamiento a partir de una base de datos de observaciones previamente cuantizadas. El objetivo principal perseguido con el entrenamiento es la estimación de los parámetros que definirán el HMM.
3) Una vez completado el entrenamiento del sistema, la clasificación de cada secuencia de observaciones cuantizadas se realiza como sigue:
   a) La secuencia de observaciones es decodificada de acuerdo a (1), obteniendo la probabilidad asociada a dicha secuencia.
   b) Esta probabilidad es comparada con un umbral de detección dado, P_{th}, de modo que la secuencia es considerada como perteneciente al modelo si la probabilidad supera el umbral.

$$Clase(O) = \begin{cases} \lambda & si\, P(O|Q,\lambda) \geq Pth \\ no\, \lambda & en\, otro\, caso \end{cases} \quad (2)$$

IV. Sistema de clasificación

Como se comentó con anterioridad para construir un sistema de identificación basado en modelos de Markov es necesario especificar: *(i)* las características que serán utilizadas como observaciones del modelo de Markov y *(ii)* el modelo de Markov en sí mismo (*i.e.*, estados, transiciones, etc). Los detalles de la implementación de nuestro sistema se presentan a continuación.

A. Observaciones del modelo: preprocesado

La aproximación de clasificación de tráfico presentada se basa en la identificación de flujos. Un flujo es considerado de acuerdo con [15] como el tráfico bidireccional identificado por la tupla `<IP-origen, puerto-origen, IP-destino, puerto-destino, protocolo-IP>`, donde origen y destino son intercambiables para permitir tráfico bidireccional. Debido a que el objetivo de nuestra detección son los protocolos de aplicación, sólo nos centraremos en paquetes con información de alto nivel.

Cada flujo es descrito como una secuencia de paquetes y cada uno de estos paquetes es una observación de nuestro sistema. Antes de introducir las observaciones a nuestro sistema éstas pasan por un preprocesado compuesto de tres módulos: *(i)* parametrización, *(ii)* normalización y *(iii)* cuantización vectorial.

Parametrización. La elección del vector de características es clave e influye directamente en los resultados de detección del sistema.

En primer lugar, es deseable que las características seleccionadas sean independientes del modo de transmisión de la información. Esto permite la detección de flujos tanto encriptados como no encriptados. En segundo lugar, las características deben ser lo más representativas posible para describir el servicio con mayores garantías de éxito en referencia a la posterior detección. Teniendo esto en consideración, las características seleccionadas son:

1) *Tiempo entre llegadas (itime):* se define para un paquete como la diferencia, en segundos, entre el tiempo de llegada de éste y la llegada del paquete previo en ese flujo.
2) *Tamaño del contenido (psize):* es el tamaño, en *bytes*, de la información que porta un paquete. De este tamaño se excluyen las cabeceras correspondientes a las capas inferiores a la de detección.
3) *Cambio de dirección (chdir):* esta característica toma valor '1 ' para un paquete que viaja en sentido inverso al del paquete previo en el flujo y '-1' en el caso contrario.

Tanto el *tiempo entre llegadas* como el *tamaño del contenido* son dos características comúnmente utilizadas en la clasificación de tráfico. Pero, hasta nuestro conocimiento, el *cambio de dirección* no ha sido propuesto ni utilizado nunca. Esta característica permite caracterizar tanto protocolos basados en TCP como protocolos basados en UDP, ya que no se asume la existencia de un cliente o un servidor como sucede con la característica dirección, que sí está presente en muchos

trabajos de clasificación de tráfico.

Normalización. El objetivo del módulo de normalización es unificar el rango dinámico de las características que representan un paquete. El rango seleccionado es [-1,1], con lo que el *cambio de dirección* no debe ser modificado. Con respecto al *tiempo entre llegadas*, primero realizamos una transformación logarítmica para reducir el rango dinámico [16]. Posteriormente, se normaliza el valor resultante realizando un desplazamiento en media y un autoescalado, $v_s = (v_o - \mu)/\sigma$, donde v_s es el valor escalado, v_o el original y μ y σ la media y desviación estándar de los valores a escalar. Para el *tamaño del contenido* sólo se aplica un desplazamiento en media y un autoescalado.

Para aplicar la normalización descrita es necesario extraer previamente la media y desviación estándar de los datos. Estos valores se calculan en la fase de entrenamiento sobre los datos dispuestos al efecto.

Cuantización vectorial. Finalmente, se lleva a cabo un proceso de cuantización sobre los vectores de características obtenidos para cada observación (paquete) de un flujo dado. La cuantización vectorial utilizada en el sistema propuesto se basa en el algoritmo K-medias. De este modo, todas las posibles combinaciones de los valores de las tres características se representan por K vectores. Estos vectores son los centroides correspondientes a los K *clusters* derivados del agrupamiento de los datos contenidos en la base de datos de entrenamiento. La métrica utilizada en dicho agrupamiento es la distancia Euclídea.

En consecuencia, cada vector de características (cada paquete) quedará representado por el índice del centroide más cercano. De este modo, las observaciones consideradas en el HMM serán una secuencia de índices de centroides derivada de la cuantización vectorial de cada paquete entrante.

B. Estructura del HMM

La característica clave de la metodología de detección utilizada es el uso de un modelo de Markov para describir el tráfico correspondiente al protocolo/servicio objetivo. Aunque para cada servicio/protocolo a detectar será necesario especificar los parámetros del modelo, a continuación presentamos un modelo genérico para representar la mayoría de los servicios de comunicación:

1) *Diálogo inicial:* Representa el inicio de la comunicación, en el que los miembros de la misma intercambian un identificador que será utilizado con posterioridad. Por ejemplo, en el caso de SSH este inicio implica una solicitud de autenticación de usuario y su correspondiente respuesta.
2) *Intercambio de información:* Aquí, los participantes transmiten la información objetivo de la comunicación, transferencia de archivos, descarga de una página web, etc. Se espera que esta fase esté compuesta por la mayoría de los paquetes intercambiados.
3) *Finalización:* Después de la fase de intercambio de información es usual que algunos protocolos envíen paquetes indicando la finalización de la comunicación. Por ejemplo SSH, sigue este proceso para finalizar una comunicación.

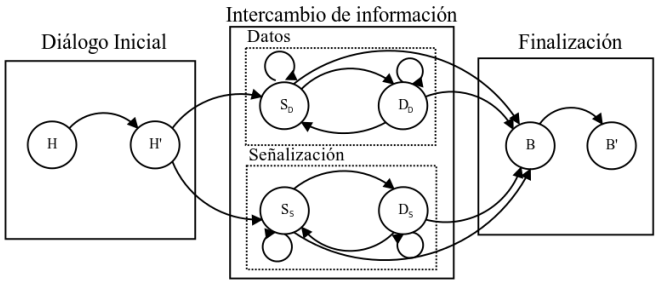

Fig. 2: HMM para el modelado del protocolo eDonkey

Un aspecto importante a tener en cuenta en el diseño de la estructura del modelo de Markov es la variabilidad en el número de observaciones (paquetes) de los flujos de un servicio/protocolo. Este hecho es abordado definiendo la etapa de *intercambio de información* como "reejecutable". De este modo, las etapas de *diálogo inicial* y *finalización* se ejecutan una única vez mientras que la de intercambio de información es ejecutada tantas veces como sea necesario.

C. HMM para la detección de los flujos eDonkey

Con el objetivo de explorar el modelo genérico propuesto en un caso de estudio, se realiza una adaptación para el caso del protocolo eDonkey.

El protocolo eDonkey presenta algunas características que lo hacen especialmente adecuado para resaltar la potencialidad de nuestra propuesta. En primer lugar, al ser un protocolo P2P, no existen clientes ni servidores. Además, este protocolo se utiliza para compartir archivos, lo que implica que existen flujos de datos y otros sólo de señalización. Y finalmente, permite dos tipos de comunicación: ofuscado y no ofuscado. Estas características implican una elevada variabilidad, lo que dificulta su modelado.

El protocolo eDonkey utiliza tanto UDP como TCP pero los flujos TCP, representan más de un 95% de los bytes transmitidos[1]. Por tanto, nuestra detección se centrará en los flujos TCP.

El modelo de Markov propuesto se puede ver en la Fig. 2. La primera y última etapa están compuestas de dos estados con una única transición entre ellos. Esto se debe a que el inicio de la comunicación en eDonkey está precedido de los mensajes `Hello` (representado por H) y `Hello answer` (H'). La finalización de la comunicación se suele componer de `Start upload request` (B) y `Queue rank` (B'). `Start upload request` se envía para solicitar la entrada en la cola de subida de un nodo destino y `Queue rank` indica la posición que se ocupa en dicha cola.

En cuanto a la etapa de *intercambio de información*, no se puede identificar un estado para cada paquete en el flujo ya que el número de paquetes por flujo es variable. En este

[1]Estos porcentajes se extrajeron de un estudio de las trazas descritas en la Sección V.

caso se definen dos caminos posibles uno para representar los flujos que nosotros denominamos de datos (número elevado de paquetes con un contenido elevado) y el otro para flujos de señalización (número reducido de paquetes en el flujo con contenido reducido). Cada uno de estos dos caminos está compuesto de dos estados, uno para representar los paquetes de datos (D) y otro para los de señalización (S). Como se mencionó con anterioridad, las transiciones en esta etapa dan cabida a tantos paquetes como se sucedan en la etapa de intercambio, pudiendo pasar a la etapa final tanto desde señalización como desde datos.

V. Evaluación experimental

En esta sección se describe la evaluación experimental realizada para validar la aproximación propuesta.

A. Bases de datos

Dos grupo de trazas de red han sido utilizadas para realizar la evaluación del sistema propuesto. Las características principales de estas trazas se exponen a continuación.

Trazas de tráfico eDonkey, eD-DB: Este conjunto está compuesto por el tráfico eDonkey generado por un nodo durante 45 días, 5 horas cada día (desde el 7 de julio al 9 de septiembre de 2011). La aplicación utilizada para compartir archivos mediante eDonkey fue aMule 2.2.6 [17].

Las trazas se componen de 240.851 flujos TCP y 7.003 UDP, todos ellos pertenecientes al protocolo eDonkey. Entre los flujos TCP 22.409 estaban ofuscados. Todas estas comunicaciones se llevaron a cabo entre más de 12.000 IPs diferentes, con una transferencia de más de 20GB de descarga y 25GB de subida.

Trazas de un troncal de una universidad de Oriente Medio, ME-DB: Este conjunto contiene todo el tráfico generado durante 48 horas (noviembre de 2010) en una universidad de Oriente Medio. En resumen, hay alrededor de 73.000 IPs y 300 millones de paquetes transmitidos.

Analizando la traza completa mediante una aplicación de inspección de contenido de los paquetes (`OpenDPI` [18]), no se encontró ningún flujo eDonkey probablemente debido a las técnicas de ofuscación de este protocolo.

Para entrenar el modelo se utiliza un subconjunto de eD-DB que sólo contiene flujos eDonkey. Para testear el sistema se usa el resto de eD-DB y un subconjunto de ME-DB. Al estar la base de datos de test etiquetada, es posible extraer las tasas de aciertos y falsos positivos. Finalmente se realiza un proceso de validación con un subconjunto de eD-DB y de ME-DB que no ha sido utilizado para entrenar ni para testear.

Adicionalmente, es importante resaltar que se lleva a cabo un proceso de la validación cruzada en la experimentación con nueve partes para entrenamiento y una para test.

B. Análisis de flujos reales de eDonkey

En primer lugar pretendemos inspeccionar si las trazas reales de eDonkey se comportan tal y como esperábamos. Para esto, se ha extraído el tipo de mensaje de los dos primeros y dos últimos paquetes de 240.851 flujos TCP de eDonkey. Se utilizó una modificación del cliente aMule para ser capaces monitorizar el contenido de todos los paquetes incluyendo los de flujos ofuscados.

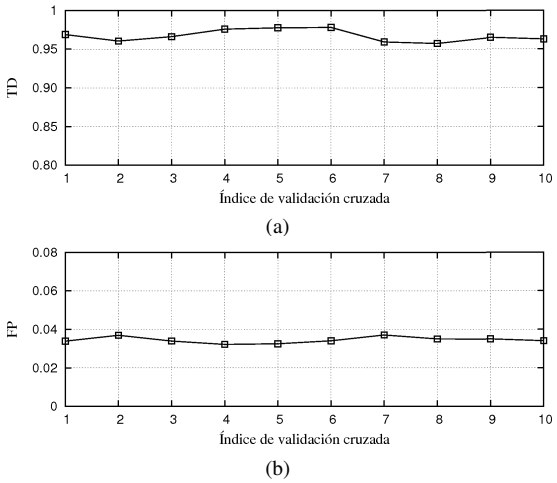

Fig. 3: TD (a) y FP (b) para el proceso de validación cruzada

Como resultado de esta inspección podemos afirmar que un 99,89% de los flujos comienzan con mensajes `HELLO` y `HELLO answer`. Adicionalmente, el 96,24% de los flujos finalizan con los mensajes `Start upload request` y `Queue rank`. Se ha comprobado también que este porcentaje no es del 100% debido a la existencia de flujos que acaban de forma abrupta.

A la luz de estos resultados se confirma que las hipótesis asumidas para definir la estructura del HMM para modelar eDonkey son aceptables.

C. Resultados de detección

El éxito del control de accesos pretendido depende de la bondad del proceso de detección, el cual se evalúa en lo subsiguiente. Para este estudio se han utilizado 32 *clusters* para el K-medias y la distancia Euclídea como métrica.

En la Fig. 3 se muestran los resultados del proceso de validación cruzada. En la subfigura (a) se puede ver la tasa de aciertos (TD) y en la (b) la tasa de falsos positivos (FP) para cada una de las 10 evaluaciones implicada en la validación cruzada. En todos los casos se obtiene más de un 95% de TD y alrededor de un 4% de FP.

Se ha realizado un estudio de la dependencia de la calidad de la detección en función de la elección del umbral de detección, P_{th} ver (2). Para ello, se ha evaluado la base de datos de validación barriendo diferentes valores de P_{th} en el rango [0,0293-0,3488]. Los resultados de este estudio se presentan como curva ROC en la Fig. 4. Como puede verse, existe un amplio rango de valores de TD (0,9 a 0,95) en los que FP es reducido (de 0,036 a 0,038). Nótese que nuestro sistema generaliza bien, ya que los resultados obtenidos en la validación son muy cercanos a los que se obtuvieron en la validación cruzada.

Como se mencionó, eD-DB contiene tanto flujos ofuscados como no ofuscados. Particularizando en la tasa de aciertos

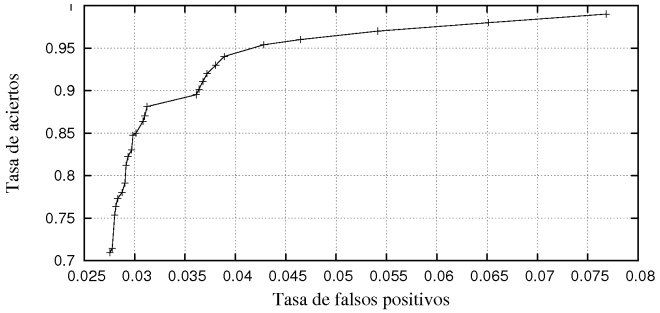

Fig. 4: Curva ROC de la detección eDonkey

TABLA I: Tasa de falsos positivos por protocolo

	Flujos totales	**FP**
HTTP	710.037	0,036
Streaming multimedia (RTP)	58.509	0,001
Compartición de archivos P2P	215.203	0,017

obtenidos en estos casos se obtuvo un TD de 96,6% para el caso de flujos ofuscados, frente a un 94,2% para los no ofuscados. Estos resultados demuestran que el sistema presentado es independiente de la existencia o no de encriptación en el protocolo objetivo.

Finalmente, se lleva a cabo una evaluación adicional que consiste en el estudio de los resultados de detección obtenidos cuando no existen flujos del protocolo eDonkey, lo que implica que todo flujo detectado será un falso positivo. Se han escogido tres grupos de servicios de ME-DB: *(i)* HTTP, *(ii)* protocolos para *streaming* multimedia (RTP) y *(iii)* protocolos para la compartición de archivos mediante P2P.

Los resultados obtenidos en esta evaluación se muestran en la Tabla I. En ésta se pueden ver las tasas de falsos positivos en función del protocolo original de los flujos evaluados. Nótese que nuestro sistema de detección es capaz de diferenciar tanto flujos de aplicaciones de compartición de archivos mediante P2P como de *streaming* multimedia con una tasa de falsos positivos muy reducida. Finalmente, HTTP es también diferenciado de eDonkey con una tasa de falsos positivos bastante aceptable.

En resumen, los resultados de detección obtenidos demuestran la bondad de nuestra aproximación en comparación con las presentes en la literatura, tal y como se discutió en la Sección II.

VI. Conclusiones y trabajo futuro

En este trabajo se introduce una metodología basada en el uso de modelos de Markov para determinar el uso de ciertos servicios/protocolos fundamentado en la clasificación de tráfico. En primer lugar, se contribuye con un HMM genérico a nivel de flujo y cuyas observaciones son los paquetes del flujo. Tras esto, se particulariza el modelo para representar las comunicaciones del protocolo eDonkey como caso de estudio. Para ello se discuten los aspectos clave involucrados: cuantización vectorial, normalización, estructura del sistema, etc.

Para validar la propuesta presentada se realiza una experimentación con tráfico eDonkey y no-eDonkey. Los resultados obtenidos muestran elevadas tasas de detección junto a reducidas tasas de falsos positivos. El sistema mantiene esta buena actuación incluso cuando los flujos de eDonkey están ofuscados o los protocolos evaluados son también de tipo P2P de compartición de archivos como BitTorrent o Gnutella.

Algunas líneas de trabajo que estamos abordando en este momento en relación al estudio aquí desarrollando son: extensión del modelado de Markov a otros servicios/protocolos y caracterización de tráfico y eventos en entornos MANET.

Agradecimientos

Este trabajo ha sido parcialmente financiado a través del MICINN mediante el proyecto TEC2011-22579.

Referencias

[1] M. Hirvonen and J.-P. Laulajainen, "Two-phased network traffic classification method for quality of service management," in *Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium on*, may 2009, pp. 962–966.

[2] J. Wenjuan and Z. Peng, "QoS routing algorithm based on traffic classification in LEO satellite networks," in *Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on*, may 2011, pp. 1–5.

[3] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proceedings of the 13th international conference on World Wide Web*, ser. WWW '04. New York, NY, USA: ACM, 2004, pp. 512–521.

[4] A. Madhukar and C. Williamson, "A Longitudinal Study of P2P Traffic Classification," in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, 2006, pp. 179–188.

[5] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A Survey on Internet Traffic Identification," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 37–52, Aug. 2009.

[6] C. Rottondi and G. Verticale, "Using packet interarrival times for Internet traffic classification," in *Communications (LATINCOM), 2011 IEEE Latin-American Conference on*, oct. 2011, pp. 1–6.

[7] F. Dehghani, N. Movahhedinia, M. Khayyambashi, and S. Kianian, "Real-Time Traffic Classification Based on Statistical and Payload Content Features," in *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*, may 2010, pp. 1–4.

[8] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '05, vol. 35, no. 4. New York, NY, USA: ACM, 2005, pp. 229–240.

[9] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian Neural Networks for Internet Traffic Classification," *Neural Networks, IEEE Transactions on*, vol. 18, no. 1, pp. 223–239, Jan. 2007.

[10] H. Dahmouni, S. Vaton, and D. Rossé, "A markovian signature-based approach to IP traffic classification," in *Proceedings of the 3rd annual ACM workshop on Mining network data*, ser. MineNet '07. New York, NY, USA: ACM, 2007, pp. 29–34.

[11] C. V. Wright, F. Monrose, and G. M. Masson, "On Inferring Application Protocol Behaviors in Encrypted Network Traffic," *J. Mach. Learn. Res.*, vol. 7, pp. 2745–2769, December 2006.

[12] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *Proceedings of the 2006 ACM CoNEXT conference*, ser. CoNEXT '06. New York, NY, USA: ACM, 2006, pp. 6:1–6:12.

[13] A. Dainotti, W. de Donato, A. Pescape, and P. Salvo Rossi, "Classification of Network Traffic via Packet-Level Hidden Markov Models," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, Nov. 2008, pp. 1–5.

[14] G. Fink, *Markov models for pattern recognition: from theory to applications*. Springer, 2008.

[15] K. Thompson, G. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *Network, IEEE*, vol. 11, no. 6, pp. 10–23, nov/dec 1997.

[16] A. Feldmann, "Characteristics of TCP Connection Arrivals," AT&T Labs Research, Technical Memorandum, 1998.

[17] aMule. Last accessed: Feb. 2012. [Online]. Available: http://www.amule.org/

[18] OpenDPI. Last accessed: Feb. 2012. [Online]. Available: http://www.opendpi.org/