UNIVERSITY OF BEIRA INTERIOR
Engineering

Classification of Peer-to-Peer Traffic by Exploring the Heterogeneity of Traffic Features Through Entropy

João Vasco Paulo Gomes

Thesis for obtaining the degree of Doctor of Philosophy in

Computer Science and Engineering

(3rd Cycle Studies)

Supervisor: Prof. Dr. Mário Marques Freire (University of Beira Interior)
Co-supervisor: Prof. Dr. Paulo Miguel Nepomuceno Pereira Monteiro (Nokia Siemens Networks Portugal)

Covilhã, March 2012

Thesis prepared at Nokia Siemens Networks Portugal, within Optical Engineering Network Optimization, and at *Instituto de Telecomunicações*, within Multimedia Signal Processing – Covilhã Group, and submitted to University of Beira Interior for defence in a public examination session.

Work financed by the Portuguese *Fundação para a Ciência e a Tecnologia* through grant contracts SFRH/BDE/15643/2006 and SFRH/BD/60654/2009 under the programme QREN – POPH – Type 4.1 — Advanced Training, co-funded by the European Social Fund and by national funds from the Portuguese *Ministério da Educação e Ciência*.

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

POPH PROGRAMA OPERACIONAL POTENCIAL HUMANO

QREN QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL PORTUGAL 2007.2013

UNIÃO EUROPEIA
Fundo Social Europeu

Dedicatory

To my family.

For all the love and support.

Acknowledgments

This thesis would not have been possible without the help of many people.

First and foremost, I would like to thank my parents for having been by my side in the hardest moments of this work, always giving me encouragement and support. But, most of all, I would like to thank them for the love they always gave me and for having taught me to always be honest and humble.

I am equally thankful to my brother and to Ana for their help and support whenever I needed it and also for putting up with me during more than a year as housemates in Lisbon.

My recognition is also due to Pedro Inácio for all the amazing discussions we had during the last years, for all the help and support, and for the long talks that always started with *"I have an idea..."*. Nevertheless, more than for anything else, I would like to thank him for his great friendship.

I am also grateful to the friends I made in the research group of Nokia Siemens Networks Portugal, not only for all the great and inspiring discussions we had as colleagues, but also for the jokes, the lunches, the roasted chestnut parties, the football matches on Mondays, and many other moments we had as friends. If there was a single reason why I would never regret of that experience, it would be because of them.

Being part of the Multimedia Signal Processing – Covilhã Group at *Instituto de Telecomunicações* and working with the talented and creative people that form that team was also a great pleasure to me. I thank my colleagues for all the support and incentive during the last year.

Last, but not least, I have to express my gratitude to my supervisor, Prof. Mário Marques Freire, for offering me the possibility of pursuing my PhD, and for the guidance, trust, and motivation, especially in the most difficult moments. I am also thankful to my co-supervisor, Prof. Paulo Monteiro, for giving me the opportunity to develop the PhD research in enterprise environment and for supervising my work at Nokia Siemens Networks Portugal.

“The Internet is the first thing that humanity has built that humanity doesn’t understand, the largest experiment in anarchy that we have ever had.”

Eric Schmidt

Foreword

This thesis describes the research work performed in the scope of the 4-year doctoral research programme and presents its conclusions and contributions. The research activities, in the initial period of the doctoral programme, were carried in the enterprise environment of Nokia Siemens Networks Portugal, through a collaboration agreement between the company and University of Beira Interior. In the latter period of the doctoral programme, the research activities were performed in the Multimedia Signal Processing – Covilhã Group of *Instituto de Telecomunicações*, at University of Beira Interior. The research work was supervised by Prof. Dr. Mário Freire, from University of Beira Interior, and co-supervised by Prof. Dr. Paulo Monteiro, at Nokia Siemens Networks Portugal. In the first period of the doctoral programme, this work was financially supported in equal parts by the Portuguese *Fundação para a Ciência e a Tecnologia* and by Nokia Siemens Networks Portugal, through the tripartite grant contract SFRH/BDE/15643/2006. In the second period, this work was financially supported by the Portuguese *Fundação para a Ciência e a Tecnologia* through the grant contract SFRH/BD/60654/2009. The research activities were also partially funded by the project TRAMANET: Traffic and Trust Management in Peer-to-Peer Networks with contracts PTDC/EIA/73072/2006 and FCOMP-01-0124-FEDER-007253.

The research work developed in enterprise environment was integrated in the activities of the research group of Siemens Networks company, which was created after the separation of the Information and Communications business division of the multinational company Siemens, in the period that preceded the establishment of the join-venture between the Networks business group of Nokia and the Information and Communications business division of Siemens. After the creation of Nokia Siemens Networks Portugal, the research group was included in the Research, Technology and Platforms business unit of the company. The group was constituted by researchers with different backgrounds, such as electrotechnical engineering, physics, or computer science, and developed research work in different topics within the field of telecommunication networks.

Conducting the doctoral research in an enterprise environment brings advantages and also a few disadvantages. Being part of a research group in a large multinational company encourages a more practical and pragmatic research approach, applied to concrete problems and oriented by the market needs and the client demands. This reality helps to reduce the gap that frequently exists between academia and industry. Furthermore, this research context facilitates the cooperation in activities and projects on different topics. Although, this sometimes made more difficult to be focused on the doctoral work and its objectives, it also helped me to acquire more flexibility and to become a more complete researcher. The cooperation in parallel activities gave me the possibility of working in different topics and with other researchers,

which resulted in one invention report related with the doctoral work and one international patent and three invention reports on other topics. Additionally, the creation of Nokia Siemens Networks Portugal motivated some progressive modifications in the structure and organization of the company, which refocused the main research field of the research group on the next generation optical networks. During this period, I also cooperated as *research and development engineer* in the Operations and Business Software business unit, integrating the Content Team of the software for telecommunication network management NSN Service Quality Manager.

In the latter period of the doctoral programme, the research work was developed in the laboratory of the Multimedia Signal Processing – Covilhã Group of *Instituto de Telecomunicações*, at University of Beira Interior. The activities of the group are included in the field of network and multimedia computing and address different topics such as networking, security, media processing, peer-to-peer networks, or cloud computing. Although the integration of a research group in academia does not offer such a direct connection to the industry as the one made possible in an enterprise environment, it gave me the possibility, in the latter period of the doctoral programme, of developing further and materializing the ideas initiated during the work in the enterprise environment. In a higher stage of maturity, those ideas were more easily published in international journals and gave origin to additional research topics.

The research work developed during the doctoral programme and described in this thesis is the consequence of the activities performed in enterprise and academia environments and, therefore, benefited from the conjunction of the advantages of both research contexts. The sense of applicability and pragmatism of the industry and the more research focused approach of academia are reflected in this thesis and in the work it presents.

List of Publications

Articles included in the thesis resulting from this 4-year doctoral research programme

1. **Detection and Classification of Peer-to-Peer Traffic: A Survey**
   João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro
   ACM Computing Surveys, accepted for publication, 2012.

2. **Source Traffic Analysis**
   João V. P. Gomes, Pedro R. M. Inácio, Blanka Lakic, Mário M. Freire, Henrique J. A. da Silva, and Paulo P. Monteiro
   ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 6, No. 3, Article 21, 23 pages, 2010.
   DOI: 10.1145/1823746.1823755

3. **Exploring Behavioral Patterns Through Entropy in Multimedia Peer-to-Peer Traffic**
   João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro
   The Computer Journal (Oxford University Press), accepted for publication, 2011.
   DOI: 10.1093/comjnl/bxr127

4. **Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties**
   João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro
   Revised version submitted for publication in an international journal, 2011.

5. **Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy**
   João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro
   Submitted for publication in an international journal, 2011.

Other publications resulting from this doctoral research programme not included in the thesis

1. **Analysis of Peer-to-Peer Traffic Using a Behavioural Method Based on Entropy**
   João V. P. Gomes, Pedro R. M. Inácio, Mário M. Freire, Manuela Pereira, and Paulo P. Monteiro
   Proceedings of the 27th IEEE International Performance Computing and Communications Conference (IPCCC 2008), Austin, TX, USA, December 7-9, 2008
   IEEE Computer Society Press, Los Alamitos, CA, ISBN: 978-1-4244-3367-4, pp. 201-208.

2. **The Nature of Peer-to-Peer Traffic**

   João V. P. Gomes, Pedro R. M. Inácio, Mário M. Freire, Manuela Pereira, and Paulo P. Monteiro

   Part 11: Measurement and P2P Traffic Characteristics

   Handbook of Peer-to-Peer Networking, Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon (Eds.) Springer, Berlin Heidelberg, 2010, ISBN: 978-0-387-09750-3, pp. 1231-1252.

Resumo

A capacidade de classificar tráfego com base na aplicação ou protocolo que o gerou é essencial para uma gestão eficaz das redes informáticas. Apesar das aplicações de Internet terem sido normalmente baseadas no paradigma *cliente-servidor*, gerando tráfego com características bem definidas e facilmente previsíveis, o aparecimento da computação *Peer-to-Peer* (P2P) conduziu a um aumento da capacidade da rede nos pontos extremos, facilitando a partilha direta de conteúdos entre utilizadores e alterando o comportamento do tráfego nas redes de fornecedores de serviços de Internet e de organizações. Nesse contexto, a capacidade de identificar a natureza do tráfego tem-se tornado cada vez mais importante. Contudo, os métodos iniciais de classificação de tráfego, baseados na associação de números de portos dos protocolos da camada de transporte a aplicações ou protocolos específicos, perderam a sua eficácia assim que diversas aplicações de Internet adotaram a utilização de números de portos aleatórios ou portos utilizados por outras aplicações. A alternativa natural foi procurar, dentro dos pacotes, conjuntos de dados que pudessem ser utilizados como uma assinatura para o tráfego da aplicação alvo. Apesar disso, esta abordagem, normalmente designada por *inspeção profunda de pacotes*, é exigente do ponto de vista dos recursos computacionais, comprometendo a sua utilização para análise de tráfego em tempo real em redes de alto débito. Para além desta limitação, algumas aplicações começaram a cifrar o tráfego impedindo a inspeção profunda de pacotes. De forma a ultrapassar estas limitações, os investigadores têm vindo a propor novas abordagens para a classificação de tráfego, por vezes chamadas de classificação *no escuro*, que são baseadas no comportamento do tráfego e não utilizam informação do *payload* dos pacotes. Apesar de terem habitualmente uma precisão menor, na generalidade dos casos, estas abordagens permitem obter um bom compromisso entre eficácia e custo computacional, sendo ainda imunes à utilização de técnicas de cifragem. Contudo, as tentativas de desenvolver métodos comportamentais mais eficazes têm conduzido a um aumento da sua complexidade.

Esta tese é focada na identificação de tráfego P2P, tendo como objetivo propor uma nova abordagem para a classificação capaz de identificar em tempo real tráfego gerado por aplicações P2P, sem recurso ao *payload* dos pacotes. O trabalho de investigação aqui descrito, após uma revisão da literatura, iniciou-se com o estudo das características do tráfego na fonte de diversas aplicações P2P e não P2P, motivado pelo facto de uma das diferenças entre os paradigmas *cliente-servidor* e P2P ser o papel duplo desempenhado pelos nós dos sistemas P2P. Ao invés de capturar dados experimentais num ponto de agregação da rede, o tráfego de cada nó individual, correndo uma única aplicação ou um conjunto pré-definido de aplicações, foi capturado junto à sua ligação à rede. Desta forma, foi possível assegurar que o tráfego analisado foi gerado pelas aplicações estudadas e que as suas características não foram afetadas pela agregação de diferentes tipos de tráfego. O estudo incluiu a análise estatística das seguintes propriedades

do tráfego: o número de *bytes* por unidade de tempo, o tempo entre chegada de pacotes, e o comprimento do pacote.

A observação do tráfego na fonte mostrou que os comprimentos dos pacotes apresentam características distintas conforme sejam gerados por aplicações P2P ou não P2P. O tráfego de aplicações não P2P resulta, habitualmente, de conexões com um comportamento estável, sendo normalmente constituído por pacotes pequenos e grandes utilizados para enviar pedidos e confirmações e para receber conteúdos, respetivamente. Nestes casos, tanto os pacotes pequenos como os grandes apresentam habitualmente comprimentos muito homogéneos. O tráfego P2P, pelo contrário, é extremamente heterogéneo em termos de comprimentos dos pacotes, uma vez que resulta da agregação de conexões concorrentes para diversos pares. Para além disso, os mecanismos distribuídos de procura e as respostas aos pedidos de outros pares geram um grande número de pacotes pequenos com diversos comprimentos. Assim, foi desenvolvido um estudo mais aprofundado focado exclusivamente nas características dos comprimentos dos pacotes e incluindo um leque de aplicações alargado. A entropia foi utilizada para medir a heterogeneidade dos comprimentos dos pacotes e os resultados obtidos mostraram ser possível distinguir os dois tipos de tráfego. De forma a melhorar os resultados, a entropia também foi calculada utilizando gamas de 200 *bytes*. Todos os comprimentos incluídos na mesma gama foram usados no cálculo da entropia como tratando-se de comprimentos idênticos. Utilizando esta abordagem, foi possível propor um novo classificador comportamental capaz de identificar nós que estejam a correr aplicações P2P, sem que seja necessário utilizar informações do *payload* dos pacotes. De forma a tornar o método aplicável em análises em tempo real, a entropia é calculada recorrendo a uma janela deslizante com um tamanho constante de *N* pacotes.

Apesar do método de classificação proposto ter a capacidade de identificar nós correndo aplicações P2P através da análise da heterogeneidade dos comprimentos dos pacotes do tráfego agregado de cada nó, não consegue ainda assim classificar fluxos individuais como tendo sido gerados por aplicações P2P ou não P2P. Na verdade, a heterogeneidade dos comprimentos dos pacotes observada no tráfego de cada nó individual que esteja a correr aplicações P2P para partilha de ficheiros ou para *media streaming* resulta sobretudo da agregação de várias conexões com propriedades distintas, utilizadas para partilhar conteúdos com outros pares. Desta forma, a heterogeneidade em fluxos individuais é menor, mesmo no tráfego P2P. Contudo, no caso do tráfego de aplicações P2P para *Voice over Internet Protocol* (VoIP), a heterogeneidade dos comprimentos dos pacotes resulta da utilização de *codecs* de voz com *bit rate* variável, sendo a heterogeneidade, por essa razão, observável nos fluxos individuais utilizados para transportar os dados referentes a sessões VoIP.

Dessa forma, foi recolhido tráfego experimental gerado por aplicações P2P para VoIP utilizando diferentes *codecs* de voz com *bit rate* variável e constante e utilizado para estudar os comprimentos dos pacotes gerados por sessões VoIP. Os resultados da análise mostraram que os

comprimentos dos pacotes são dependentes do *codec* de voz utilizado na sessão. Assim, a heterogeneidade dos comprimentos dos pacotes de cada sessão foi medida utilizando a entropia, calculada com recurso a uma janela deslizante com um tamanho constante de 500 pacotes. Para cada *codec* de voz considerado no estudo, foram compilados os intervalos de comprimentos dos pacotes e entropia observados durante a análise de tráfego, tendo sido proposto um classificador, baseado nesses intervalos, capaz de identificar tráfego VoIP recorrendo a uma única propriedade do tráfego. O classificador faz uso de um conjunto de assinaturas comportamentais associadas a cada *codec* de voz, constituídas por um intervalo de comprimentos de pacotes e um intervalo da entropia dos comprimentos dos pacotes. Para além de reconhecer tráfego VoIP *no escuro*, o classificador consegue ainda identificar o *codec* de voz utilizado na respetiva sessão VoIP.

Após a proposta do classificador de tráfego P2P VoIP, o trabalho de investigação focou-se no tráfego gerado por aplicações P2P para partilha de ficheiros ou para *media streaming*. Ao contrário do VoIP, o tráfego gerado por um único nó de rede a correr uma aplicação deste tipo resulta de várias conexões paralelas para diversos pares. Por essa razão, nesta tese, o tráfego P2P para partilha de ficheiros ou para *media streaming* é designado por tráfego P2P *um para vários*. A entropia dos comprimentos dos pacotes de fluxos individuais destas aplicações não é suficientemente diferençável da entropia obtida para fluxos individuais de aplicações não P2P. Assim, foram estudadas separadamente diversas dimensões do tráfego, incluindo pacotes de entrada, de saída, e de saída e entrada em conjunto, e também pacotes cujo comprimento é menor ou igual a 100 *bytes*, maior que 100 *bytes* e menor ou igual a 900 *bytes*, ou maior que 900 *bytes*. A média da entropia dos comprimentos dos pacotes em cada uma destas dimensões foi calculada para cada fluxo das aplicações analisadas, utilizando uma janela deslizante com um tamanho constante de 100 pacotes. Para além disso, foi ainda calculada a média da entropia dos tempos entre chegada de pacotes e dos pares *endereço*/*porto* remotos com os quais cada par *endereço*/*porto* local comunica. Com base nos resultados obtidos, foi proposto um classificador de tráfego que não recorre ao *payload* dos pacotes. Durante a avaliação do desempenho, o classificador demonstrou ser capaz de identificar tráfego P2P com uma precisão de 95%.

Palavras-chave

Classificador Comportamental, Classificação de Tráfego, Classificação no Escuro, *Codecs* de Voz, Comportamento do Tráfego de Redes, Comprimentos de Pacotes, Entropia, Inspeção de Tráfego, *Media Streaming*, Monitorização e Análise de Tráfego, Partilha de Ficheiros, *Peer-to-Peer* (P2P),

Tráfego Multimédia, *Voice over Internet Protocol* (VoIP).

Resumo Alargado

Introdução

Este capítulo resume, de forma alargada e em Língua Portuguesa, o trabalho de investigação descrito na tese de doutoramento intitulada "*Classification of Peer-to-Peer Traffic by Exploring the Heterogeneity of Traffic Features Through Entropy*". A parte inicial deste capítulo descreve o enquadramento da tese, define o problema abordado e os objetivos do doutoramento, apresenta o argumento da tese, e descreve as suas principais contribuições. De seguida, é abordado o tópico de investigação sobre a classificação de tráfego em redes informáticas e são apresentados com maior detalhe os trabalhos de investigação e as principais contribuições da tese. O capítulo termina com a discussão breve das principais conclusões e a apresentação de algumas linhas de investigação futura.

Enquadramento da Tese

A classificação de tráfego tem-se tornado num recurso importante na gestão das redes informáticas. A capacidade de identificar a aplicação ou o protocolo que gera o tráfego é essencial para grande parte das tarefas relacionadas com a administração de redes informáticas, tais como a gestão eficaz do tráfego, a definição de políticas de qualidade de serviço e de medidas de segurança, ou a correta configuração de redes informáticas [1]. Os métodos iniciais para classificação de tráfego eram baseados na associação de números de portos dos protocolos da camada de transporte a protocolos de aplicação específicos, como por exemplo, os portos 80 e 21 habitualmente usados, respetivamente, pelos protocolos *Hypertext Transfer Portocol* (HTTP) e *File Transfer Protocol* (FTP). Estes métodos são facilmente implementados em sistemas de monitorização de tráfego e a sua eficácia era grande quando o tráfego de rede mantinha um comportamento previsível e as aplicações utilizavam portos bem definidos.

Até ao final do século passado, as aplicações de Internet baseavam-se sobretudo no paradigma *cliente-servidor*, tendo, cada nó de rede, bem definida a função única de *cliente* ou de *servidor*. Neste contexto, os clientes faziam pedidos aos servidores e estes forneciam-lhes serviços ou conteúdos. Desta forma, a carga de tráfego nas ligações de Internet era geralmente assimétrica, apresentando um fluxo de tráfego superior do servidor para o cliente. Contudo, o aparecimento do paradigma de computação *Peer-to-Peer* (P2P) aumentou capacidade da rede nos pontos extremos, oferecendo aos nós a possibilidade de partilharem conteúdos e serviços diretamente entre eles, desempenhando, assim, um papel duplo de *cliente* e *servidor*. Como

consequência, o tráfego nos pontos extremos aumentou substancialmente ao mesmo tempo que a assimetria da carga de tráfego foi reduzida. A largura de banda usada na distribuição de conteúdos e normalmente suportada por servidores dedicados, passou a ser partilhada por fornecedores de serviços de Internet e redes locais de organizações [2]. Além disso, a partilha de conteúdos diretamente entre utilizadores, facilitada pelo paradigma P2P, proporcionou também o aumento da propagação de vírus, *cavalos de tróia*, e outras ameaças [3]. Também a privacidade, o anonimato, e a confidencialidade são aspetos sensíveis que podem ser ameaçados pela natureza distribuída das aplicações P2P [4].

Assim, os fornecedores de serviços de Internet e os administradores de redes de organizações passaram a limitar, ou mesmo bloquear, o tráfego gerado por aplicações P2P, por forma a evitarem ou a controlarem os efeitos destas aplicações nas suas redes. De maneira a evitarem a identificação do seu tráfego, as aplicações P2P começaram a usar números de portos aleatórios. Mais tarde, quando os mecanismos de classificação de tráfego passaram a recorrer à inspeção profunda de pacotes para melhorarem a sua precisão [5], várias aplicações P2P começaram a cifrar o *payload* dos seus pacotes, inviabilizando assim a sua inspeção. Para além disso, os métodos de classificação de tráfego baseados na inspeção profunda dos pacotes são habitualmente considerados mais exigentes em termos computacionais. Tipicamente, estes métodos procuram a identificação, no conteúdo de cada pacote, de assinaturas de *bytes* associadas a cada aplicação alvo. Uma vez que a lista de assinaturas é por vezes longa, a inspeção profunda de pacotes em tempo real em redes de alto débito poderá ser difícil. A inspeção do *payload* dos pacotes do tráfego poderá ainda levantar questões legais de violação de privacidade [6].

Em alternativa à classificação de tráfego baseada nos números de portos e na inspeção profunda de pacotes, diversos estudos têm vindo a propor novos métodos que recorrem à análise das características comportamentais do tráfego, sendo por vezes chamados de métodos de classificação *no escuro* [7]. A maioria dos métodos comportamentais baseia-se na análise estatística de características do tráfego ou na utilização de heurísticas que possam modelar o comportamento de aplicações ou protocolos [8]. Uma vez que não recorrem à identificação, no *payload* dos pacotes, de assinaturas específicas de cada aplicação, a eficácia destes métodos é geralmente menor quando comparada com a eficácia dos métodos baseados na inspeção profunda de pacotes, sendo, no entanto, imune à cifragem de pacotes. Apesar de, normalmente, a classificação *no escuro* ser computacionalmente menos exigente, a tentativa de melhorar a eficácia dos métodos baseados no comportamento do tráfego tem vindo a aumentar a complexidade das soluções propostas em diversos estudos. Cascarano *et al.* [9] compararam o custo computacional de um classificador comportamental baseado em *Support Vector Machines* (SVMs) e de um classificador baseado na inspeção profunda de pacotes, concluindo que ambos os classificadores têm custos computacionais semelhantes. Para além das diferenças na eficácia, o facto de os métodos comportamentais não se basearem em assinaturas específicas para cada aplicação,

mas sim nas propriedades genéricas de cada classe de tráfego, permite-lhes classificar tráfego da classe alvo gerado por aplicações ou protocolos emergentes ou desconhecidos.

O âmbito desta tese está limitado às áreas da monitorização e análise de tráfego e das redes P2P. O trabalho de investigação aqui descrito foca-se nos desafios para a gestão de tráfego levantados pelo tráfego de aplicações P2P e nas diferentes abordagens para classificação de tráfego. O desenvolvimento de métodos de classificação que não sejam afetados pela cifragem do tráfego e que sejam capazes de processar o tráfego em tempo real é motivado pelas limitações das abordagens baseadas na inspeção profunda de pacotes e pela busca por métodos de classificação que não façam uso do *payload* dos pacotes. Os métodos de classificação propostos nesta tese são baseados na análise da heterogeneidade de características do tráfego, sobretudo do comprimento dos pacotes.

Descrição do Problema e Objetivos da Investigação

O problema abordado nesta tese de doutoramento é o da classificação de tráfego em tempo real, sem recurso à informação contida no *payload* dos pacotes. Motivados pelas limitações dos métodos baseados nos números de portos [10] ou na inspeção profunda de pacotes [11], diversos estudos têm proposto novos métodos de classificação baseados no comportamento do tráfego. Contudo a generalidade desses métodos apresenta também diversas limitações, normalmente relacionadas com a incapacidade de serem aplicados na monitorização em tempo real [12], com o facto de serem focados num número reduzido de aplicações [13] ou de exigirem uma fase de aprendizagem [14], ou com a utilização de algoritmos complexos e a análise de um número elevado de características do tráfego [15].

Neste contexto, o objetivo principal desta tese consiste na proposta de um novo método de classificação de tráfego capaz de classificar tráfego gerado por aplicações P2P, sem estar limitado a aplicações ou protocolos específicos. O método proposto deverá ser capaz de classificar o tráfego sem recurso ao *payload* dos pacotes, de forma a que possa ser utilizado para classificar tráfego cifrado. O classificador deverá ainda ser capaz de classificar o tráfego em tempo real, sendo capaz de produzir resultados durante a duração dos fluxos. Para além disso, deverá ser utilizado um número de características do tráfego reduzido, sendo evitados métodos complexos cujas exigências computacionais cresçam exponencialmente com a quantidade de tráfego analisada.

O trabalho de investigação necessário para cumprir o objetivo da tese foi estruturado nos seguintes objetivos intermédios:

1. De forma a conhecer os métodos e abordagens existentes, deverá ser feita uma revisão

do estado da arte no tópico da classificação de tráfego, sendo analisadas as vantagens e limitações de cada abordagem. Deverão ainda ser estudados os desafios criados pelas aplicações P2P na gestão de tráfego e as maiores dificuldades que os métodos de classificação enfrentam na identificação deste tipo de tráfego.

2. Como base para a identificação de propriedades comportamentais que pudessem ser utilizadas na classificação de tráfego, é objetivo desta tese estudar o tráfego de diferentes aplicações P2P e não P2P capturado diretamente na fonte, imediatamente após o nó que o gerou. Desta forma, é possível garantir que as propriedades observadas no tráfego são consequência da respetiva aplicação e não da agregação do tráfego de várias fontes.

3. A partir do estudo do objetivo anterior, pretende-se identificar propriedades do tráfego que permitam propor um método comportamental de classificação de tráfego capaz de distinguir o tráfego de nós de rede que estejam a correr aplicações P2P, sem recurso ao *payload* dos pacotes.

4. O tráfego de um nó de rede pode resultar da agregação do tráfego de diversas aplicações que correm no nó, sendo também, por essa razão, objetivo desta tese classificar fluxos individuais de tráfego P2P. Uma vez que, ao contrário do que acontece com as aplicações P2P para *media streaming* ou para partilha de ficheiros, cada aplicação P2P para *Voice over Internet Protocol* (VoIP) gera apenas um fluxo de tráfego, o método proposto para classificação de fluxos deverá considerar ambos os casos.

Argumento da Tese

Esta tese propõe uma nova abordagem para a classificação de tráfego P2P. O argumento apresentado nesta tese é o seguinte:

A natureza distribuída do paradigma P2P influencia propriedades do tráfego, como os comprimentos dos pacotes, aumentando a sua heterogeneidade. O nível de heterogeneidade dessas propriedades pode ser medido usando a entropia e aplicado na caracterização do tráfego P2P. A análise da entropia dessas propriedades do tráfego pode ser utilizada na classificação em tempo real de tráfego P2P, sem recurso ao payload *dos pacotes.*

De forma a sustentar este argumento, foi utilizada a seguinte abordagem.

São estudados o problema e a área de investigação e analisadas as vantagens e limitações das diferentes abordagens para classificação de tráfego.

As propriedades do tráfego de aplicações P2P e não P2P, capturado na fonte, são analisadas, com especial atenção para o número de *bytes* por unidade de tempo, o tempo entre chegada de

pacotes, e o comprimento dos pacotes. Este estudo permite identificar características distintas nos dois tipos de tráfego, para serem utilizadas na classificação de tráfego.

São recolhidas amostras de tráfego mais longas e mais variadas e a entropia é usada para medir a heterogeneidade identificada nos comprimentos dos pacotes. De forma a calcular a entropia em tempo real, é utilizada uma janela deslizante contendo um número constante de pacotes. O cálculo da entropia com recurso à janela deslizante é utilizada em todos os classificadores propostos nesta tese.

De forma a mostrar a viabilidade da utilização da heterogeneidade dos comprimentos dos pacotes, medida através da entropia, para distinguir tráfego P2P e não P2P, é proposto um classificador para classificar o tráfego de cada nó de rede como contendo ou não tráfego gerado por aplicações P2P. O classificador proposto usa quatro regras baseadas na entropia dos comprimentos dos pacotes sob três perspetivas distintas.

Após a apresentação deste classificador, é abordada a classificação individual dos fluxos. Primeiro, são estudados os comprimentos dos pacotes de amostras de tráfego VoIP gerado com diversos *codecs* de voz e aplicações e é analisada a sua entropia. Os valores de entropia observados são usados para criar assinaturas comportamentais de cada *codec*, aplicadas num novo classificador para tráfego VoIP. Depois, é analisada a entropia dos comprimentos dos pacotes segundo diferentes perspetivas baseadas em intervalos de comprimentos, juntamente com a entropia dos tempos entre chegada de pacotes e dos pares *endereço*/*porto* remotos. Os resultados da média da entropia desde a primeira iteração da janela, para cada perspectiva analisada, são usados na definição de regras de um novo classificador para a classificação individual de fluxos gerados por aplicações P2P.

Principais Contribuições

A primeira contribuição desta tese é a análise das abordagens e métodos existentes para classificação de tráfego e das suas vantagens e limitações, assim como a revisão abrangente da literatura no tópico da classificação de tráfego P2P. Este estudo está descrito com detalhe no capítulo 2, que consiste num artigo aceite para publicação na revista *ACM Computing Surveys* [16].

A segunda contribuição desta tese é o estudo das características do tráfego capturado na fonte e gerado por diversas aplicações multimédia P2P e não P2P, tendo sido analisadas propriedades como o número de *bytes* por unidade de tempo, o tempo entre chegada de pacotes, ou o comprimento do pacote. A análise do tráfego na fonte permitiu identificar propriedades no tráfego que resultassem da aplicação e não da agregação do tráfego de diversas fontes. O

estudo desenvolvido está descrito no capítulo 3, que consiste num artigo publicado na revista *ACM Transactions on Multimedia Computing Communications and Applications* [17].

A terceira contribuição desta tese é a observação e o estudo da heterogeneidade dos comprimentos dos pacotes no tráfego gerado por aplicações P2P e não P2P, a apresentação de um método aplicável em tempo real que permita quantificar essa heterogeneidade utilizando a entropia, e a proposta de um novo classificador baseado nos nós de rede. O método proposto recorre a uma janela deslizante com um tamanho fixo de *N* pacotes, permitindo assim que o cálculo da entropia seja efetuado imediatamente a partir do *N*-ésimo pacote analisado. A versão inicial deste estudo foi publicada nas atas duma conferência internacional [18], tendo as principais conclusões sido também descritas num capítulo de um livro [19]. A análise de tráfego foi melhorada de forma a incluir um conjunto alargado de aplicações e um novo classificador capaz de classificar o tráfego de nós de rede que corram aplicações P2P foi apresentado. O método de classificação proposto não utiliza informação do *payload* dos pacotes, sendo baseado na entropia dos seus comprimentos. De forma a poder operar em tempo real, o classificador usa o mecanismo de janela deslizante descrito anteriormente. O estudo da heterogeneidade dos comprimentos dos pacotes e o classificador proposto estão descritos com detalhe no capítulo 4, que consiste num artigo aceite para publicação na revista *The Computer Journal* [20].

A quarta contribuição desta tese é o estudo dos comprimentos dos pacotes do tráfego gerado por sessões VoIP utilizando diferentes aplicações P2P e *codecs* de voz e a apresentação de um classificador de tráfego VoIP capaz de identificar o *codec* de voz utilizado em casa sessão. A análise de tráfego experimental de cada *codec* e aplicação permitiu identificar intervalos de comprimentos dos pacotes e correspondente entropia. Estes intervalos foram depois utilizados para formar diversas assinaturas comportamentais que integraram um classificador de tráfego apresentado, capaz de classificar tráfego de sessões VoIP e de identificar o *codec* de voz utilizado na sessão. O estudo e o classificador são descritos no capítulo 5, que consiste na versão revista de um artigo submetido para publicação numa revista internacional [21].

A quinta e última contribuição desta tese é a proposta de um classificador de tráfego capaz de identificar os fluxos individuais do tráfego gerado por aplicações P2P *um para vários*. O classificador é baseado, sobretudo, na entropia dos comprimentos dos pacotes, fazendo também uso da análise da entropia dos tempos entre chegada de pacotes e dos pares *endereço*/*porto* remotos com que cada par *endereço*/*porto* local comunica. O cálculo da entropia foi feito recorrendo a uma janela deslizante, permitindo assim a sua utilização em tempo real. Em cada iteração da janela, foi calculada a média da entropia desde a primeira iteração. Os resultados da média da entropia obtidos na análise de tráfego foram utilizados na definição de um conjunto de regras utilizado pelo classificador para identificar fluxos P2P. Este trabalho é descrito com maior detalhe no capítulo 6, que consiste num artigo submetido para publicação numa revista internacional [22].

Classificação de Tráfego em Redes Informáticas

O trabalho de investigação apresentado nesta tese inclui o estudo das abordagens existentes para classificação de tráfego, dos seus pontos fortes e limitações, e dos contextos em que a utilização de cada uma delas oferecerá maiores vantagens. Esse estudo detalhado está descrito no capítulo 2, juntamente com uma introdução a diversos conceitos base na monitorização de tráfego, como são a captura correta de tráfego, os diferentes níveis de observação dos dados recolhidos na rede, ou a redução do tráfego recolhido por via da análise de amostras. Após a introdução à área de estudo, o capítulo apresenta uma revisão abrangente da literatura sobre o tópico da classificação de tráfego, com especial incidência nas publicações que se focam na classificação do tráfego gerado por aplicações P2P.

O método inicial para classificação de tráfego consistia na associação de números de portos dos protocolos da camada de transporte a aplicações ou protocolos de aplicação específicos, como por exemplo, os portos 80 ou 21 associados com os protocolos HTTP ou FTP. Apesar da classificação do tráfego baseada nos números dos portos ser facilmente implementada em sistemas de monitorização de tráfego, a sua eficácia ficou bastante reduzida quando um grande número de aplicações, especialmente as que pretendem disfarçar o seu tráfego, passaram a utilizar números de portos aleatórios. Os estudos publicados onde eram utilizados os números de portos para a classificação de tráfego tinham como objetivo, essencialmente, a análise da carga de tráfego P2P nas redes de computadores [10, 23]. A maioria destes trabalhos foi publicada numa altura em que a utilização de números de portos aleatórios por parte das aplicações não era, ainda, comum. A tabela II do capítulo 2 apresenta os portos que eram, habitualmente, utilizados por algumas aplicações P2P.

A classificação de tráfego baseada na inspeção profunda de pacotes consiste, normalmente, na utilização de uma lista de assinaturas formadas por *strings* de dados específicas habitualmente encontradas no *payload* dos pacotes de determinadas aplicações ou protocolos [5, 24]. Em cada pacote processado, o classificador verifica se o conteúdo do pacote corresponde a uma das regras. No caso dessa correspondência se verificar, o pacote é classificado como tendo sido gerado pela aplicação associada à regra. Este processo é, habitualmente, exigente do ponto de vista dos recursos computacionais. Uma vez que estes métodos se baseiam no *payload* dos pacotes, a sua utilização fica limitada nos casos em que o tráfego é cifrado. Com o objetivo de identificarem tráfego cifrado, alguns investigadores exploraram, também, a aleatoriedade dos *bytes* no *payload* dos pacotes introduzida pela cifragem [13, 25].

Com o objetivo de evitarem as limitações dos métodos baseados na inspeção profunda de pacotes, vários estudos têm vindo a propor novos classificadores baseados no comportamento do tráfego. Este tipo de abordagem, por vezes chamada de classificação de tráfego no escuro [7, 26],

baseia-se na utilização de características genéricas do tráfego e no recurso a diferentes métodos para as processar: heurísticas [27], relações entre pares [7], assinaturas estatísticas [28], algoritmos de aprendizagem automática [29], ou a identificação de serviços [30]. A tabela V do capítulo 2 resume os artigos publicados que propõem novos métodos comportamentais para classificação de tráfego, assim como os resultados da avaliação do desempenho efetuada pelos seus autores.

Alguns autores propuseram, também, métodos ativos que simulam o comportamento normal da aplicação alvo da classificação [31], tomando a iniciativa de estabelecer ligações com outros pares que corram a mesma aplicação, e identificando assim o tráfego alvo. A combinação de diferentes abordagens foi também proposta por alguns estudos na literatura [13].

Análise do Tráfego na Fonte

Uma das fases iniciais do trabalho de investigação descrito nesta tese consistiu no estudo das características do tráfego de aplicações P2P e não P2P. Os resultados desse estudo, descrito no capítulo 3, permitiram identificar diferenças nas propriedades do tráfego dos dois tipos de aplicação que seriam depois exploradas de forma a construir classificadores de tráfego P2P. Em vez de usar tráfego capturado num ponto de agregação, o trabalho desenvolvido baseou-se na análise do tráfego capturado na sua fonte, junto ao ponto onde foi gerado. Desta forma, é possível garantir que as propriedades observadas resultam da aplicação sob estudo e não da agregação de diferentes fontes de tráfego.

O trabalho iniciou-se com um estudo breve do trabalho desenvolvido por outros autores na área da simulação e modelação de tráfego de rede. Algumas das propriedades do tráfego de aplicações da Internet em pontos de agregação da rede, descritas em alguns trabalhos da literatura, são apresentadas na subsecção 2.1 do capítulo 3. A partir dessa descrição, são também apresentadas, na subsecção 2.2 do capítulo 3, as propriedades do tráfego VoIP, vídeo, e dados, conforme foram descritas em estudos disponíveis na literatura.

A análise do tráfego experimental baseou-se nas amostras apresentadas na tabela VI do capítulo 3. As propriedades do número de *bytes* por unidade de tempo, do tempo entre chegada de pacotes, e do comprimento do pacote foram estudadas e diferentes distribuições conhecidas foram utilizadas para modelar os dados experimentais. Os valores empíricos do número de *bytes* por unidade de tempo foram modelados pela distribuição de *Weibull* ou pela distribuição *Normal*, enquanto que no caso do tempo entre chegada de pacotes, os dados empíricos foram modelados, na generalidade, pela distribuição de *Weibull*.

Os resultados mais interessantes para o objetivo desta tese, foram, no entanto, os obtidos para o comprimento dos pacotes. A maioria dos valores observados no tráfego de aplicações não P2P formavam uma distribuição bimodal ou trimodal, tendo os comprimentos dos pacotes apenas dois ou três valores diferentes. Os valores empíricos dos comprimentos dos pacotes gerados por aplicações P2P, por sua vez, foram modelados, sobretudo, pela distribuição de *Weibull*, já que apresentavam diversos valores distintos. Esta observação foi o ponto de partida para as atividades de investigação que se seguiram. As distribuições usadas para modelar cada tipo de tráfego são resumidas na tabela VII do capítulo 3.

Heterogeneidade dos Comprimentos dos Pacotes do Tráfego P2P

A análise de tráfego seguinte focou-se exclusivamente no comprimento dos pacotes, de forma a explorar os indícios observados no estudo anterior. O tráfego experimental capturado e analisado foi aumentado e o conjunto de aplicações consideradas na análise foi alargado, conforme apresentado na subsecção 3.1 do capítulo 4.

Os resultados da análise dos pacotes mostraram que os seus comprimentos eram bastante mais variados no tráfego gerado por aplicações P2P. Essa heterogeneidade resulta da agregação de múltiplos fluxos que cada par estabelece com outros pares para a partilha de conteúdos. No caso das aplicações P2P para VoIP, a maior variedade de comprimentos de pacotes deve-se aos *codecs* de *bit rate* variável usados por aplicações como o *Skype* ou o *Google Talk* na maior parte das situações. Por forma a medir essa heterogeneidade e a transformá-la numa quantidade utilizável num classificador de tráfego, foi usado o conceito de entropia definido por Shannon na teoria da informação [32]. A entropia, habitualmente denotada por $H(x)$, foi calculada usando a expressão (1), onde n representa o número de valores de x para os quais a estatística é calculada, e $p(x_i)$ é a probabilidade da ocorrência específica do valor de x_i:

$$H(x) = -\sum_{i=1}^{n} p(x_i) \ln p(x_i). \tag{1}$$

Para qualquer número finito $n \in \mathbb{N}$, o valor máximo de $H(x)$ é dado por

$$H(x) = \ln n. \tag{2}$$

Uma vez que a entropia tem que ser calculada para um número finito de valores, por forma a obter o seu valor em tempo real, foi utilizada uma janela deslizante, contendo um número constante de comprimentos de pacotes. Cada comprimento de pacote analisado é colocado na janela deslizante até esta estar cheia. Assim que a janela é preenchida pela primeira vez, o

valor da entropia é calculado para os comprimentos dos pacotes que estão incluídos na janela. Quando um novo comprimento de pacote é processado, o valor que está há mais tempo na janela sai e entra o comprimento de pacote mais recente. Desta forma, a janela mantém o seu tamanho constante, criando um movimento virtual sobre os comprimentos dos pacotes analisados. Em cada iteração da janela, um novo valor de entropia é calculado, possibilitando assim a análise da heterogeneidade dos comprimentos dos pacotes em tempo real.

Contudo, calcular repetidamente o valor da entropia em cada iteração da janela, implica o cálculo das probabilidades de todos os valores, o que pode tornar-se numa tarefa computacionalmente exigente. Por forma a tornar o processo de cálculo menos exigente, a entropia é calculada apenas na momento em que a janela é preenchida pela primeira vez. A partir daí, o peso na entropia do comprimento do pacote que sai da janela é atualizado no valor da entropia, assim como o peso do comprimento do pacote que entra na janela. A atualização do peso de um comprimento *c* na iteração *i* da janela foi calculado usando a expressão (3), sendo o cálculo da entropia na mesma iteração automatizado usando a expressão (4), onde c_s e c_e são, respetivamente, os comprimentos dos pacotes que saem e entram na janela:

$$U(c) = p_{i-1}(c) \ln p_{i-1}(c) - p_i(c) \ln p_i(c), \tag{3}$$

$$H_i(x) = H_{i-1}(x) + U(c_s) + U(c_e). \tag{4}$$

O tamanho da janela deslizante tem influência nos valores máximos obtidos, que crescem quando o tamanho da janela aumenta. No entanto, para tamanhos superiores a 100 pacotes, a entropia aumenta apenas ligeiramente quando o tamanho da janela aumenta. O efeito mais interessante do aumento do tamanho da janela é a maior estabilidade do valor da entropia, como é exemplificado pela figura 4 do capítulo 4. Assim, é necessário em cada análise escolher um tamanho que ofereça o melhor compromisso entre a estabilidade do valor da entropia e o tempo necessário para encher a janela e obter o primeiro valor de entropia.

Os valores obtidos para a entropia dos comprimentos dos pacotes são bastante distintos para tráfego P2P e não P2P, tal como é demonstrado na tabela 1 do capítulo 4. As aplicações P2P geram o tráfego com maior entropia dos comprimentos dos pacotes, apresentando as aplicações P2P para *media streaming* ou para partilha de ficheiros comprimentos de pacotes também variados mas com menor entropia. As aplicações não P2P geram pacotes com comprimentos muito homogéneos, apresentando por isso um nível baixo de entropia. Para maximizar as diferenças entre os dois tipos de tráfego, foi também analisada a entropia excluindo os pacotes contendo apenas mensagens de *acknowledgement*. Estes pacotes não resultam da natureza da aplicação, mas sim do protocolo da camada de transporte *Transmission Control Protocol* (TCP). Uma vez que todos estes pacotes têm comprimentos semelhantes e são utilizados pelos dois tipos de aplicações, considerá-los na análise aproxima mais as características dos pacotes, ao invés de

as diferenciar. Este estudo de tráfego e o método proposto de análise da entropia em tempo real são as bases dos classificadores propostos a seguir.

Classificação do Tráfego de Nós de Rede Correndo Aplicações P2P

Os padrões de heterogeneidade do comprimento dos pacotes identificados durante a análise do tráfego experimental resultam, sobretudo, da agregação de diferentes fluxos de tráfego. Assim, é possível desenvolver um classificador de tráfego P2P que se baseie nos valores de entropia observados para os diferentes tipos de aplicação. A maior dificuldade dessa abordagem seria o facto de o mesmo utilizador, normalmente, correr mais que uma aplicação ao mesmo tempo. Por essa razão, o efeito que a agregação do tráfego de diferentes aplicações tem no resultado final da entropia para um nó de rede foi estudado no âmbito deste trabalho.

A preocupação inicial seria verificar se a agregação do tráfego de aplicações P2P e não P2P correndo no mesmo nó de rede resultaria ainda assim num valor de entropia elevado, ou se a presença do tráfego não P2P baixaria consideravelmente a entropia, não permitindo detetar a presença do tráfego P2P. Foram, por isso, analisadas amostras de tráfego de utilizadores correndo, paralelamente, aplicações P2P e não P2P, sendo visível que, apesar da presença de tráfego não P2P, o valor da entropia mantém-se ainda assim elevado, conforme se mostra na figura 5 do capítulo 4.

Seria ainda necessário verificar se a agregação do tráfego de várias aplicações não P2P resultaria num valor de entropia elevado, podendo, assim, dar a ideia incorreta de que se trataria de tráfego P2P. Portanto, foram capturadas amostras de tráfego de utilizadores correndo um grande número de aplicações multimédia não P2P. Para extremar a análise e tentar obter valores de entropia mais elevados, o número de aplicações utilizadas em paralelo por cada utilizador foi exagerado para além do espectável para um utilizador normal, podendo incluir, por exemplo, *streaming* de um grande número de conteúdos vídeo e áudio em paralelo juntamente com outras aplicações não P2P. Apesar de, na maior parte dos casos, a entropia ser baixa, houve alguns exemplos para os quais a entropia subiu para valores semelhantes aos observados para tráfego P2P. Para distinguir estes casos, foi analisada a entropia apenas para tráfego de saída, e também utilizando intervalos de 200 *bytes* para o cálculo das probabilidades. Todos os valores incluídos num desses intervalos eram usados no cálculo da entropia como se se tratasse do mesmo valor. Desta forma, foi possível distinguir a presença e a ausência de tráfego P2P no tráfego agregado de um nó de rede, mesmo quando um número exagerado de aplicações não P2P eram utilizadas em paralelo. A tabela 2 do capítulo 4 resume os resultados obtidos.

Utilizando os resultados obtidos, foi proposto um classificador capaz de classificar o tráfego de

nós de rede que corram aplicações P2P. O classificador utiliza apenas quatro regras baseadas na entropia dos comprimentos dos pacotes no tráfego geral, no tráfego de saída, e no tráfego de saída usando intervalos de 200 *bytes*. Durante a avaliação do desempenho, a taxa de falsos positivos variou entre 0.00% e 10.42%, enquanto que a taxa de falsos negativos variou entre 7.69% e 12.50%.

Classificação de Fluxos Individuais de Tráfego P2P

A heterogeneidade dos comprimentos dos pacotes explorada no classificador apresentado anteriormente resulta, sobretudo, da agregação dos vários fluxos de tráfego gerados pelas aplicações P2P. Contudo, cada sessão de uma aplicação P2P para VoIP gera um fluxo de tráfego cuja heterogeneidade do comprimento dos pacotes depende do *codec* de voz utilizado. Aplicações como o *Skype* ou o *Google Talk* usam, na maioria dos casos, *codecs* de *bit rate* variável, gerando pacotes com comprimentos extremamente heterogéneos. No entanto, podem também ser utilizados *codecs* de *bit rate* constante, gerando assim pacotes com comprimentos muito semelhantes. Já no caso das aplicações P2P para *media streaming* ou para partilha de ficheiros, a entropia dos comprimentos dos pacotes para cada fluxo individual poderá não ser suficiente para distinguir, de forma clara, o tráfego P2P e não P2P. Como a heterogeneidade dos comprimentos dos pacotes tem causas distintas nos casos do tráfego de aplicações P2P para VoIP e aplicações P2P para *media streaming* ou para partilha de ficheiros, os dois casos foram abordados de formas diferentes.

Classificação do Tráfego de Sessões VoIP com Base nas Propriedades do *Codec*

Com o objetivo de estudar os comprimentos dos pacotes gerados por aplicações P2P para VoIP, foram capturadas amostras de tráfego de diversas sessões VoIP utilizando várias aplicações e *codecs* de voz, com *bit rate* variável e constante, conforme apresentado na tabela 1 do capítulo 5. A heterogeneidade dos comprimentos dos pacotes foi medida usando o método, já descrito, baseado na entropia, tendo sido compilados, para cada *codec* de voz, os intervalos em que os valores observados para os comprimentos dos pacotes e a respetiva entropia estavam contidos. Esses intervalos foram utilizados para formar uma lista de assinaturas comportamentais, apresentadas na tabela 2 do capítulo 5 e incluídas num classificador de tráfego de sessões VoIP.

O classificador proposto analisa os comprimentos dos pacotes por fluxo e calcula a sua entropia. Para cada fluxo, é utilizada uma janela deslizante independente. Como as sessões VoIP mantêm características semelhantes nos dois sentidos do tráfego, os pacotes de entrada e de saída são

analisados separadamente para despistar casos de outros tipos de tráfego que apresentassem características semelhantes a uma das assinaturas apenas num dos sentidos. Os valores obtidos são comparados com as assinaturas na lista e, caso os valores estejam contidos nos intervalos de uma das assinaturas, o fluxo é classificado como sendo uma sessão VoIP em que o *codec* associado à assinatura foi utilizado. Como a entropia é calculada com recurso a uma janela deslizante, o método proposto pode ser utilizado na classificação de tráfego em tempo real, apresentando, assim, resultados desde o início do fluxo até ao seu final. Também o facto do classificador apresentado ser baseado apenas nos comprimentos dos pacotes e na sua entropia evita a utilização de informação do *payload* dos pacotes. Por essa razão, o método proposto pode ser usado na classificação de tráfego cifrado.

O desempenho do classificador foi avaliado num ambiente laboratorial, juntamente com tráfego de outros tipos de aplicações de maneira a testar a robustez da classificação. Desta forma, foi também possível ter a certeza da verdadeira natureza do tráfego, sem a necessidade de confiar na eficácia dum classificador alternativo que funcionasse como base de comparação. Uma vez que o tráfego VoIP, em termos de número de fluxos, está em inferioridade no conjunto do tráfego de todas as aplicações do ambiente de teste, foram utilizadas como métricas a sensibilidade (*sensitivity*) e a especificidade (*specificity*), definidas e explicadas na subsecção 4.6 do capítulo 2. Durante a avaliação do desempenho do classificador proposto, a sensibilidade variou entre 70.00% e 100.00%, enquanto que a especificidade variou entre 84.85% e 100.00%, conforme as amostras de tráfego e o grupo de assinaturas usados. Os resultados detalhados são apresentados na tabela 3 do capítulo 5. Para além da eficácia da classificação, foram também avaliados os recursos computacionais utilizados pelo classificador e a forma como estes evoluem quando a quantidade de tráfego analisada aumenta. Os resultados obtidos permitiram verificar que a percentagem do tempo de processamento requerido pelo classificador aumenta linearmente com o número de pacotes analisados, enquanto que a memória utilizada na classificação aumenta linearmente com o número de fluxos processados.

Classificação do Tráfego de Aplicações P2P de *Um Para Vários*

No caso do tráfego gerado por aplicações P2P para *media streaming* ou para partilha de ficheiros, a diferença, entre tráfego P2P e não P2P, nos valores da entropia para cada fluxo é menos evidente. Assim, a análise da entropia de forma idêntica à usada na classificação de tráfego P2P VoIP é insuficiente para classificação de fluxos individuais das aplicações P2P *um para vários*. Por essa razão, os comprimentos dos pacotes em tráfego experimental das aplicações descritas na subsecção 3.1 do capítulo 6 foram analisados separadamente sob diferentes perspetivas: pacotes cujo comprimento é menor ou igual a 100 *bytes*, maior que 100 *bytes* e menor ou igual a 900 *bytes*, e maior que 900 *bytes*. Esta análise foi feita de forma independente para todo o

tráfego, para tráfego apenas de saída, ou para tráfego apenas de entrada. Foram ainda estudadas, como características adicionais, a entropia dos tempos entre chegada de pacotes e dos pares *endereço*/*porto* remotos com os quais cada par local comunica.

Os valores de entropia identificados no estudo do tráfego foram utilizados na definição dum conjunto de regras incluído num classificador de tráfego capaz de classificar os fluxos individuais gerados por aplicações P2P. O novo classificador proposto utiliza uma janela deslizante independente para cada uma das perspetivas, conforme é mostrado na figura 11 do capítulo 6. O método de análise usado, por calcular a entropia com recurso a uma janela deslizante, pode ser aplicado à classificação em tempo real. Uma vez que não é usada informação do *payload* dos pacotes, o tráfego cifrado é igualmente identificado pelo classificador apresentado.

O classificador foi avaliado num ambiente laboratorial de teste, semelhante ao utilizado para avaliar o classificador de tráfego VoIP, tendo sido utilizadas as métricas exatidão (*accuracy*), precisão (*precision*), e revocação (*recall*), definidas e explicadas na subsecção 4.6 do capítulo 2. Os resultados obtidos na avaliação do desempenho variaram entre 95.68% e 96.49% para a exatidão, entre 97.52% e 98.94% para a precisão, e entre 92.90% e 94.50% para a revocação. As tabelas 3 e 4 do capítulo 6 apresentam, com maior detalhe, os resultados obtidos. A análise dos recursos computacionais utilizados pelo classificador mostra que o tempo de processador utilizado pelo classificador cresce linearmente com o números de pacotes processados, enquanto a memória usada aumenta linearmente com o número de pares *endereço*/*porto* analisados, conforme é ilustrado pela figura 12 do capítulo 6.

Principais Conclusões

Esta tese é focada na classificação de tráfego gerado por aplicações P2P, descrevendo o trabalho de investigação desenvolvido com o propósito de propor um novo método de classificação capaz de classificar tráfego P2P em tempo real e sem recurso ao *payload* dos pacotes. Os métodos atuais apresentam limitações, sendo, por essa razão, objetivo desta tese que o classificador proposto superasse algumas dessas limitações, como a complexidade do método ou o número de características do tráfego utilizadas na classificação.

Assim, durante o trabalho de investigação desenvolvido no âmbito desta tese de doutoramento, foram estudados os problemas e desafios criados pelas aplicações P2P, do ponto de vista da gestão das redes de informação e do seu tráfego. Esses desafios têm motivado o desenvolvimento de novos métodos de classificação de tráfego utilizando diferentes abordagens. Os métodos baseados na inspeção profunda de pacotes têm normalmente uma maior eficácia, sendo, no entanto, mais exigentes em termos computacionais e afetados pela cifragem do *payload* dos

pacotes. Por outro lado, os métodos baseados no comportamento do tráfego, apesar de apresentarem, geralmente, uma menor eficácia, oferecem um bom compromisso entre eficácia e custo computacional, sendo ainda imunes à cifragem do tráfego. O estudo detalhado das diferentes abordagens e a análise abrangente da literatura no tópico da classificação de tráfego P2P foram apresentados e discutidos nesta tese.

O trabalho de investigação apresentado nesta tese inclui a análise do tráfego capturado na fonte, gerado por várias aplicações P2P e não P2P. Diversas características do tráfego foram analisadas e modeladas utilizando distribuições conhecidas. A natureza distribuída das aplicações P2P revelou-se em algumas das características analisadas, sendo as diferenças para as aplicações *cliente-servidor* especialmente visíveis no comprimento dos pacotes gerados pelos dois tipos de aplicações. Este estudo foi a base para a maioria das contribuições desta tese, tendo influenciado o trabalho de investigação que se seguiu.

Seguindo as conclusões retiradas do trabalho anterior, o estudo do tráfego focou-se na heterogeneidade revelada pelos comprimentos dos pacotes das aplicações P2P. O tráfego capturado foi analisado separadamente para cada nó de rede e a heterogeneidade dos pacotes foi medida recorrendo à entropia. Utilizando um método baseado numa janela deslizante contendo um número constante de pacotes, foi possível implementar o cálculo da entropia em tempo real. Os resultados obtidos mostraram que a entropia do comprimento dos pacotes é bastante mais elevada no tráfego de aplicações P2P. Assim, com base nos padrões observados, foi proposto um classificador de tráfego capaz de identificar o tráfego de nós de rede que corram aplicações P2P. O método de classificação apresentado funciona em tempo real e não utiliza qualquer informação do *payload* dos pacotes.

A classificação de fluxos individuais de tráfego P2P foi abordada separadamente para tráfego de aplicações P2P para VoIP e de aplicações P2P para *media streaming* ou para partilha de ficheiros. Uma vez que os comprimentos dos pacotes gerados por sessões VoIP dependem do *codec* de voz utilizado, foi capturado tráfego experimental gerado por diversas sessões VoIP onde foram utilizados aplicações e *codecs* de voz distintos. Para cada *codec* analisado, foram compilados os intervalos onde os comprimentos dos pacotes e a respetiva entropia estavam contidos. Esses intervalos foram utilizados para formar assinaturas comportamentais para cada *codec* de voz, que foram depois integradas num classificador de tráfego capaz de classificar tráfego gerado por sessões VoIP e de identificar o *codec* de voz utilizado na sessão. O método proposto baseia-se apenas na análise dos comprimentos dos pacotes, não necessitando de recorrer à informação do *payload* dos pacotes. De forma a ser aplicado em tempo real, o classificador utiliza o esquema de janela deslizante já descrito anteriormente.

No caso das aplicações P2P para *media streaming* ou para partilha de ficheiros, a heterogeneidade resulta sobretudo da agregação de diferentes fluxos, sendo as diferenças na entropia

menos óbvias quando se analisa individualmente cada fluxo. Assim, a análise da heterogeneidade dos pacotes com recurso à entropia foi feita separadamente para pacotes de entrada e de saída e também para os pacotes com comprimento pertencente a cada um de três intervalos diferentes de comprimentos de pacotes: menores ou iguais a 100 *bytes*, de 101 a 900 *bytes*, e maiores que 900 *bytes*. Para além disso, de forma a melhorar os resultados, foi ainda analisada a heterogeneidade dos tempos entre chegada de pacotes e dos pares *endereço*/*porto* remotos com que cada par *endereço*/*porto* analisado comunica, através do cálculo da entropia para estas características do tráfego. Os resultados obtidos foram utilizados para criar regras utilizadas por um método apresentado para a classificação de fluxos individuais de tráfego P2P. De forma a operar em tempo real, o classificador proposto faz uso de uma janela deslizante para cada característica do tráfego analisada. Uma vez que utiliza apenas características comportamentais do tráfego, o método proposto não recorre ao *payload* dos pacotes.

As diferentes análises de tráfego desenvolvidas ao longo do trabalho de investigação descrito nesta tese demonstram que a natureza distribuída do paradigma P2P se reflete nas características do tráfego, cujo comportamento se revela mais *caótico* e menos previsível. Estas propriedades são especialmente visíveis nos comprimentos dos pacotes gerados por estas aplicações, apresentando uma maior heterogeneidade quando comparados com os comprimentos dos pacotes gerados por aplicações *cliente-servidor*. A contribuição desta tese para explorar essa heterogeneidade com recurso à entropia é a base das restantes contribuições aqui apresentadas. As diferenças na entropia, sobretudo, dos comprimentos dos pacotes, mas também dos tempos entre chegada de pacotes e dos pares *endereço*/*porto* remotos permitiram distinguir entre o tráfego P2P e não P2P. Recorrendo a uma janela deslizante contendo um número constante de pacotes, foi possível automatizar o cálculo da entropia em tempo real e utilizá-lo na classificação de tráfego.

O objetivo principal desta tese foi cumprido através da apresentação de três classificadores de tráfego baseados na análise da heterogeneidade de características do tráfego com recurso à entropia e na sua análise através da utilização de uma janela deslizante. Em conjunto, os três classificadores permitiram a classificação de tráfego de nós de rede correndo aplicações P2P e de fluxos individuais do tráfego gerado por aplicações P2P para VoIP e aplicações P2P para *media streaming* ou para partilha de ficheiros. Os métodos propostos minimizaram o número de características do tráfego necessárias, recorrendo sobretudo ao comprimento dos pacotes e, num dos casos, também aos tempos entre chegada de pacotes e aos pares *endereço*/*porto* remotos. Não necessitando de informação do *payload* dos pacotes, os classificadores propostos podem então ser utilizados para a classificação de tráfego cifrado. O facto dos classificadores propostos classificarem tráfego P2P genérico, sem se focarem numa aplicação ou protocolo específico, permite-lhes identificar tráfego de protocolos P2P desconhecidos ou emergentes.

Direções Para Trabalho Futuro

Uma das linhas de investigação que poderá ser desenvolvida no futuro será a cooperação entre métodos de classificação baseados na inspeção profunda dos pacotes e a abordagem aqui apresentada, baseada na análise da heterogeneidade de características do tráfego com recurso à entropia. Um classificador baseado na entropia de características do tráfego podia ser usado nos casos em que a inspeção profunda dos pacotes não obtivesse qualquer resultado, quer fosse por via da cifragem do *payload* dos pacotes ou devido à inexistência duma assinatura para a aplicação. Poderia, também, ser utilizado para confirmar a classificação devolvida pelo módulo de inspeção profunda de pacotes nos casos em que fosse comum aplicações P2P disfarçarem o seu tráfego de forma a ser classificado como tráfego de outra aplicação, ou em casos como, por exemplo, o da aplicação *Gnutella* que usa o protocolo HTTP, sendo por vezes o seu tráfego incorretamente classificado apenas como tráfego Hypertext Transfer Protocol (HTTP) ou *Web*.

Por forma a implementar uma abordagem de cooperação que seguisse esta descrição, seria necessário desenvolver um estudo sobre os tipos de tráfego que são geralmente classificados de forma incorreta pelos métodos que usam a inspeção profunda de pacotes e sobre quais os ganhos que seriam obtidos em termos de eficácia. Ainda assim, este esquema de cooperação não permitiria obter um ganho na eficiência.

Uma das principais razões para os métodos baseados na inspeção profunda de pacotes serem computacionalmente exigentes é o facto de, para um número considerável de assinaturas, terem que verificar se as *strings* de dados que formam cada assinatura são identificadas no *payload* dos pacotes. A utilização, como primeiro classificador, de um método baseado na entropia poderia permitir, dependendo do resultado de entropia obtido, verificar apenas uma parte das assinaturas. Para isso, seria necessário desenvolver um estudo sobre os níveis de entropia observados para diversas características do tráfego de diferentes tipos de aplicações. Desta forma, a lista de assinaturas do classificador baseado na inspeção profunda de pacotes poderia ser dividida em partes menores às quais seria atribuído um determinado nível de entropia. Apenas a parte das assinaturas correspondente ao nível de entropia devolvido pelo primeiro classificador seria verificada pela inspeção profunda de pacotes. Os ganhos de eficiência alcançados com este esquema de cooperação deveriam ser avaliados, de forma a serem retiradas conclusões.

A análise da entropia de características específicas do tráfego poderá também ser usada na classificação de tráfego com base apenas no nível de entropia, ao invés de classificá-lo com base na aplicação que o gerou. Uma abordagem de classificação deste tipo ajudaria a caracterizar o tráfego, podendo ser vista como um fator adicional utilizado na sua caracterização. Potencialmente, uma classificação baseada apenas na observação da entropia poderá ser usada como um fator a considerar, por exemplo, na definição de políticas de gestão de tráfego ou de esquemas

de encaminhamento. Apesar dos benefícios desta abordagem não serem à partida garantidos, seria interessante estudar a informação que poderia ser extraída do tráfego através deste tipo de análise e de que forma essa informação poderia ser aplicada em diferentes domínios da administração de redes de informação.

Referências

[1] A. Kind, X. Dimitropoulos, S. Denazis, and B. Claise, "Advanced network monitoring brings life to the awareness plane," *IEEE Commun. Mag.*, vol. 46, no. 10, pp. 140–146, Oct. 2008.

[2] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet service providers fear peer-assisted content distribution?" in *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2005)*, Berkeley, CA, USA, Oct. 2005, pp. 63–76.

[3] D. Chopra, H. Schulzrinne, E. Marocco, and E. Ivov, "Peer-to-peer overlays for real-time communication: Security issues and solutions," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 4–12, Jan.–Mar. 2009.

[4] G. Lawton, "Is peer-to-peer secure enough for corporate use?" *IEEE Computer*, vol. 37, no. 1, pp. 22–25, Jan. 2004.

[5] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2005, vol. 3431, pp. 41–54.

[6] P. Ohm, D. C. Sicker, and D. Grunwald, "Legal issues surrounding monitoring during network research," in *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2007)*, San Diego, CA, USA, Oct. 2007, pp. 141–148.

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

[8] J. Hurley, E. Garcia-Palacios, and S. Sezer, "Host-based P2P flow identification and use in real-time," *ACM Trans. Web*, vol. 5, pp. 1–27, May 2011.

[9] N. Cascarano, A. Este, F. Gringoli, F. Risso, and L. Salgarelli, "An experimental evaluation of the computational cost of a DPI traffic classifier," in *Proc. IEEE Global Communications Conf. (GLOBECOM 2009)*, Honolulu, HI, USA, Nov.–Dec. 2009, pp. 1–8.

[10] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, Apr. 2004.

[11] L. Bin, L. Zhi-Tang, and T. Hao, "A methodology for P2P traffic measurement using application signature work-in-progress," in *Proc. 2nd Int. Conf. Scalable Information Systems (InfoScale '07)*, Suzhou, China, Jun. 2007, pp. 1–2.

[12] B. Li, M. Ma, and Z. Jin, "A VoIP traffic identification scheme based on host and flow behavior analysis," *J. Netw. Syst. Manag.*, vol. 19, no. 1, pp. 111–129, Mar. 2011.

[13] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: When randomness plays with you," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Oct. 2007.

[14] M. Dusi, A. Este, F. Gringoli, and L. Salgarelli, "Coarse classification of internet traffic aggregates," in *Proc. IEEE Int. Conf. Communications (ICC 2010)*, Cape Town, South Africa, May 2010, pp. 1–6.

[15] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 50–60, Jun. 2005.

[16] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Detection and classification of peer-to-peer traffic: A survey," *ACM Computing Surveys*, accepted for publication.

[17] J. V. P. Gomes, P. R. M. Inácio, B. Lakic, M. M. Freire, H. J. A. da Silva, and P. P. Monteiro, "Source traffic analysis," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 3, pp. 1–23, Aug. 2010.

[18] J. V. P. Gomes, P. R. M. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "Analysis of peer-to-peer traffic using a behavioural method based on entropy," in *Proc. 27th IEEE Int. Performance Computing and Communications Conf. (IPCCC 2008)*, Austin, TX, USA, Dec. 2008, pp. 201–208.

[19] J. V. P. Gomes, P. R. M. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "The nature of peer-to-peer traffic," in *Handbook of Peer-to-Peer Networking*, X. Shen, H. Yu, J. Buford, and M. Akon, Eds. Berlin Heidelberg: Springer, 2010, pp. 1231–1252.

[20] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Exploring behavioral patterns through entropy in multimedia peer-to-peer traffic," *The Computer J.*, accepted for publication.

[21] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Identification of peer-to-peer VoIP sessions using entropy and codec properties," submitted for publication.

[22] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Classification of one-to-many peer-to-peer traffic using packet length and entropy," submitted for publication.

[23] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen, "P2P, the gorilla in the cable," in *Proc. National Cable & Telecommunications Association (NCTA)*, Chicago, IL, USA, Jun. 2003, pp. 8–11.

[24] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proc. 13th Int. Conf. World Wide Web (WWW '04)*, New York, NY, USA, May 2004, pp. 512–521.

[25] R. Dhamankar and R. King, "Protocol identification via statistical analysis (PISA)," *White Paper, Tipping Point*, 2007.

[26] W. H. Turkett, A. V. Karode, and E. W. Fulp, "In-the-dark network traffic classification using support vector machines," in *Proc. 20th National Conf. Innovative Applications of Artificial Intelligence (IAAI '08)*, Chicago, IL, USA, Jul. 2008, pp. 1745–1750.

[27] F. Constantinou and P. Mavrommatis, "Identifying known and unknown peer-to-peer traffic," in *Proc. IEEE Int. Symp. Network Computing and Applications (NCA '06)*, Cambridge, MA, USA, Jul. 2006, pp. 93–102.

[28] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting," *Elsevier Computer Netw.*, vol. 53, no. 1, pp. 81–97, Jan. 2009.

[29] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Elsevier Computer Netw.*, vol. 53, no. 14, pp. 2476–2490, Sep. 2009.

[30] M. Baldi, A. Baldini, N. Cascarano, and F. Risso, "Service-based traffic classification: Principles and validation," in *Proc. IEEE Sarnoff Symp. (SARNOFF '09)*, Princeton, NJ, USA, Mar.–Apr. 2009, pp. 115–120.

[31] S. Ohzahata, Y. Hagiwara, M. Terada, and K. Kawashima, "A traffic identification method and evaluations for a pure P2P application," in *Proc. Passive and Active Measurement Conf. (PAM 2005)*, ser. Lecture Notes in Computer Science, vol. 3431, Boston, MA, USA, Mar. 2005, pp. 55–68.

[32] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, Jul. 1948.

Abstract

The ability to classify the traffic, based on the application or protocol that generated it, is essential for the effective management of computer networks. Although Internet applications were generally based on the *client-server* paradigm, generating traffic whose properties were well defined and easily predictable, the advent of peer-to-peer (P2P) computing brought the power to the edges, facilitating the direct exchange of contents between hosts and modifying the behavior of the traffic load in the networks of Internet Service Providers (ISPs) and organizations. In that context, the ability to identify the nature of the traffic became increasingly important. Nonetheless, the early traffic classification methods, based on the association of port numbers of transport-level protocols to applications or protocols, became ineffective when many Internet applications started to use random port numbers or ports normally used by other applications. The natural alternative was to look deep into the contents of the packets to search for data strings that could be used as a signature of the traffic of a target application. However, this approach, usually called Deep Packet Inspection (DPI), requires more computational resources which may make it difficult to be used for real-time monitoring in high-speed networks. Moreover, several applications have started to encrypt their traffic preventing the use of DPI. In order to overcome these limitations, researchers are proposing new classification approaches, sometimes called classification *in the dark*, which are based on the traffic behavior and do not rely on the payload data. Although their accuracy is generally lower, in most cases, they offer a good compromise between effectiveness and computational cost and are not affected by encryption techniques. Nevertheless, the search for more accurate behavioral methods is also leading to an increase in their complexity.

This thesis is focused on the identification of P2P traffic and aims to propose a classification approach capable of identifying traffic generated by P2P applications in real-time, without relying on the payload data. Since one of the differences between *client-server* and P2P paradigms is the dual role played by P2P hosts, the research work described herein, after a literature review, started with the study of the properties of the traffic from several P2P and non-P2P applications at its source. Instead of collecting the experimental data in an aggregation point, the traffic from each individual host, running a single application or a predefined set of applications, was captured immediately after its network connection. By doing so, it was possible to assure that the analyzed traffic was generated by the studied applications and that its properties were not affected by the aggregation of different types of traffic. The study included the statistical analysis of the following traffic features: the byte count per time unit, the inter-arrival time, and the packet length.

The observation of the source traffic showed that the lengths of the packets generated by P2P

and non-P2P applications present distinct patterns. The traffic from non-P2P applications usually results from connections with a stable behavior, mostly formed by small and large packets, used to send requests and acknowledgments and to receive contents, respectively. In these cases, both small and large packets generally present very homogeneous lengths. On the contrary, the P2P traffic is very heterogeneous in terms of packet lengths, as it results from the aggregation of several concurrent connections to different peers. Moreover, the distributed search mechanisms and the replies to requests from other peers also generate a large number of small packets with multiple lengths. Hence, a deeper study focused solely on packet length properties was performed and the set of analyzed applications was extended. The entropy was used to measure the heterogeneity of the packet lengths and the results showed it was possible to differentiate both kinds of traffic. To improve the results in specific cases, the entropy of the packet lengths was also computed using slots of 200 bytes, which means that all the packet lengths within the same slot are used in the entropy computation as being similar lengths. Based on this approach, it was possible to propose a new behavioral classifier capable of identifying hosts running P2P applications, without using payload data. In order to make the method suitable for real-time analysis, the entropy is computed using a sliding window with a constant size of *N* packets.

Although the proposed classification method was able to identify hosts running P2P applications by analyzing the heterogeneity of the packet lengths in the aggregated traffic of each host, it could not classify individual flows as being generated by P2P or non-P2P applications. In fact, the heterogeneity of the packet lengths observed in the traffic of each single host running P2P file-sharing or P2P media streaming applications resulted, mostly, from the aggregation of several connections with different properties, used to share contents with other peers. For this reason, the heterogeneity of individual flows is lower, even for P2P traffic. Nonetheless, in the case of P2P Voice over Internet Protocol (VoIP) traffic, the heterogeneity of the packet lengths results from the use of Variable Bit Rate (VBR) speech codecs and, thus, the heterogeneity is observable in the individual flow used to carry each VoIP session.

Therefore, experimental traffic generated by P2P VoIP applications using several VBR and Constant Bit Rate (CBR) speech codecs was collected and used to study the lengths of the packets generated by VoIP sessions. The results of the analysis showed that the packet lengths depend on the speech codec used in each the session. Hence, the heterogeneity of the packet lengths from each VoIP session was measured using entropy, which was computed using a sliding window with a constant size of 500 packets. For each speech codec considered in the study, the intervals of packet lengths and entropy observed during the traffic analysis were compiled and, based on those intervals, a traffic classifier capable of identifying VoIP traffic using a single traffic feature was proposed. The classifier uses a set of behavioral signatures associated with each speech codec, formed by an interval of packet lengths and an interval of the entropy of

the packet lengths. Besides of being able to recognize VoIP traffic *in the dark*, the classifier is also capable of identifying the speech codec used in that VoIP session.

After proposing the P2P VoIP traffic classifier, the research work focused on the traffic from P2P file-sharing and P2P media streaming applications. Unlike VoIP, the traffic generated by a single host running one of these applications results from many parallel connections with several peers. Hence, in this thesis, P2P file-sharing and P2P media streaming traffic is also designated by *one-to-many P2P traffic*. The entropy of the packet lengths of individual flows from these applications is not sufficiently distinct from the entropy obtained from non-P2P individual flows. Therefore, several dimensions of the traffic were separately studied, including incoming, outgoing, or incoming and outgoing packets together, and also packets whose payload length is smaller or equal to 100 bytes, greater than 100 bytes and smaller or equal to 900 bytes, or greater than 900 bytes. The mean of the entropy of the packet lengths for each of these dimensions was computed for each flow of the analyzed applications, using a sliding window with a constant size of 100 packets. Additionally, the mean of the entropy of the inter-arrival times and of the remote *host*/*port* pairs to which a local *host*/*port* pair communicates was also computed. Based on the obtained results, a traffic classifier that does not rely on payload data was proposed. In the performance evaluation, the classifier was able to identify P2P traffic with an accuracy greater than 95%.

Keywords

Behavioral Classifier, Classification in the Dark, Entropy, File-sharing, Media Streaming, Multimedia Traffic, Network Traffic Behavior, Packet Lengths, Peer-to-Peer (P2P), Speech Codecs, Traffic Classification, Traffic Inspection, Traffic Monitoring and Analysis, Voice over Internet Protocol (VoIP).

Contents

List of Figures

Chapter 4

Exploring Behavioral Patterns Through Entropy in Multimedia Peer-to-Peer Traffic

Chapter 5

Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties

List of Tables

Chapter 5

Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties

A Supplement to "Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties"

Chapter 6

Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy

A Supplement to "Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy"

Acronyms

ACF	AutoCorrelation Function
ADSL	Asymmetric Digital Subscriber Line
AIM	AOL Instant Messenger
AMR-WB	Adaptive Multi-Rate Wideband
ART	Adaptive Resonance Theory
ATM	Asynchronous Transfer Mode
CART	Classification And Regression Tree
CBR	Constant Bit Rate
CI	Confidence Interval
CMP	chip multiprocessor
CVFDT	Concept-adapting Very Fast Decision Tree
DAG	Data Acquisition and Generation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DFA	Deterministic Finite Automata
DHT	Distributed Hash Table
DNS	Domain Name System
DPI	Deep Packet Inspection
DSL	Digital Subscriber Line
EG711	Enhanced *G.711*
EM	Expectation-Maximization
FARIMA	Fractional AutoRegressive Integrated Moving Average
FN	False Negative
FP	False Positive
FPGA	Field Programmable Gate Array
FTP	File Transfer Protocol
GIPS	Global IP Solutions
GMM	Gaussian Mixture Model
GSM	Global System for Mobile communications
GTVS	Ground Truth Verification System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
iLBC	Internet Low Bit Rate Codec
IMAP	Internet Message Access Protocol

IMRG	Internet Measurement Research Group
IP	Internet Protocol
iPCMwb	Internet Pulse Code Modulation wideband
IPFIX	Internet Protocol Flow Information eXport
IRC	Internet Relay Chat
IRTF	Internet Research Task Force
iSAC	Internet Speech Audio Codec
ISP	Internet Service Provider
ITU	International Telecommunication Union
LAN	Local Area Network
MAC	Media Access Control
MB	mediumband
ML	Machine Learning
MMS	Microsoft Media Server
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NB	narrowband
NIDS	Network Intrusion Detection System
OS	Operating System
P2P	peer-to-peer
P2PTV	peer-to-peer television
PART	Partial Decision Tree
PCM	Pulse-Code Modulation
POP	Post Office Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAM	Random Access Memory
RPE-LTP	Regular Pulse Excitation Long-Term Prediction
RQA	Recurrence Quantification Analysis
RTCP	Real-Time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
SB-ADPCM	Sub-Band Adaptive Differential Pulse Code Modulation
SFTP	Secure File Transfer Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
SVM	Support Vector Machine

SVOPC	Sinusoidal Voice Over Packet Coder
TCAM	Ternary Content Addressable Memory
TCP	Transmission Control Protocol
TDG	Traffic Dispersion Graph
TES	Transform Expand Sample
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
VBR	Variable Bit Rate
VFDT	Very Fast Decision Tree
VoD	Video on Demand
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network
WB	wideband
WWW	World Wide Web
XMPP	Extensible Messaging and Presence Protocol

Chapter 1

Introduction

This thesis addresses the subject of classification of peer-to-peer (P2P) traffic, proposing behavioral classification methods based on the heterogeneity of traffic features. The focus and scope of the thesis are further described in this chapter, together with the problem definition and objectives, the thesis statement, the main contributions, and the thesis organization.

1 Thesis Focus and Scope

Traffic classification has always been an important task in computer network administration. The capability to identify the application or protocol that generated each traffic flow is crucial not only for the effective traffic management and correct design of computer networks, but also to assure the Quality of Service (QoS) required by applications with distinct priority levels or to implement security measures for different applications [1, 2]. The early methods used to identify the traffic nature were simply based on the association of an application or protocol to a well-known port number used by the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) to transmit data. The association of, e.g., Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP) traffic width ports 80 or 21, respectively, is illustrative of the traditional use of port numbers to identify the application protocols that generated specific traffic flows.

By the end of the last century, Internet applications were mainly based on the *client-server* paradigm, in which the *client* and *server* roles played by each host in a connection were clearly defined and separated. Servers provided services and contents to clients who request them, generating an asymmetric amount of data in upstream (from client to server) and downstream (from server to client). In this scenario, where Internet applications traffic resulted from connections with conventional and easily predictable properties and using well-known port numbers, and the traffic load in computer networks followed well-studied models, the accurate classification of traffic was a fairly simple task relying on port numbers. However, more recently, the emergence of the P2P paradigm pushed the power to the network edges by offering the users the possibility of sharing contents among themselves, acting concurrently as clients and servers. Although P2P is only a computing paradigm, it raises challenges distinct from the ones associated with other network architectures. Instead of using bandwidth only to re-

ceive contents and services, P2P hosts share among them the cost of the bandwidth needed to provide the contents, which, in the *client-server* paradigm, was supported solely by the dedicated servers. As a consequence, not only the traffic in the networks of Internet Service Providers (ISPs) and organizations increased considerably, as the characteristics of the traffic load also changed. Besides having to support the traffic grow, ISP networks, which were prepared to offer larger bandwidth rates in downstream, also have to deal with a traffic increase in upstream that makes the traffic load in both directions less asymmetrical. Therefore, this shift of the power to the network edges is increasing the costs supported by networks of ISPs and organizations [3]. Furthermore, the ability to facilitate the sharing of data directly between hosts has also raised several security issues that may cause impact in companies and home users [4–6]. The consequences of virus, worms, and other threats are magnified by the multiple direct connections between hosts, easily provided by P2P applications. Privacy, anonymity, and confidentiality are also sensitive issues that may be threaten by an incautious use of P2P applications [7]. Although these problems are not a direct consequence of the P2P paradigm, the possibility it offers of directly sharing contents between users, together with the encryption and obfuscation techniques used by many P2P applications, make it more difficult to identify potential threats in P2P traffic.

Motivated by the problems raised mainly by P2P applications, ISPs and network administrators started to limit or block the traffic generated by those applications. However, the ability to identify the application that generated the traffic was heavily compromised by the adoption of random port numbers or even the use of well-known port numbers that are associated with other protocols whose traffic is, generally, less constrained by monitoring devices, e.g, port 80 [8]. In order to improve their accuracy, classification mechanisms implemented Deep Packet Inspection (DPI) methods, which resort to the data carried within the packets, in the payload field of the transport level protocols [9]. Typically, DPI classifiers use a repository of payload strings associated with the target applications or protocols and try to match each of the signatures with the payload data in each processed packet. Although they have usually a good accuracy, classification mechanisms based on payload signatures have also several limitations and their effectiveness is decreasing in specific contexts. Since they have to check every signature of the target applications in each processed packet, DPI methods generally require more computation resources and, thus, are more difficult to be used in real-time. Moreover, due to the increase of the number of protocols and Internet applications and their complexity, the number of signatures that have to be checked is growing. The deep inspection of the packet contents may also raise privacy concerns [10]. Additionally, many P2P applications are starting to encrypt the payload data, which prevents the verification of data signatures in the packet payload [11]. Nevertheless, instead of using data signatures, some authors analyzed the randomness of the payload bytes, introduced by encryption techniques, to identify encrypted traffic [12].

In order to avoid the limitations of DPI, many researchers have started to propose traffic classification methods that do not rely on payload data, sometimes called classification *in the dark* methods [13, 14]. Most of them are based on statistical measures or heuristics that use different traffic features to model the common behavior of applications and protocols [15, 16]. Some behavioral classifiers for P2P traffic use generic properties of this kind of traffic, enabling their application to the classification of P2P traffic from unknown protocols [17]. Moreover, these methods keep their effectiveness in the classification of encrypted traffic as they do not need to use payload data. Nevertheless, since behavioral classifiers only explore the generic behavior of the traffic and do not rely on any specific payload signatures, their accuracy is generally lower when compared to DPI methods. Moreover, in most cases, they can only identify the protocol instead of the specific application, e.g., they may be able to recognize traffic generated by a *BitTorrent* client, but usually they cannot identify the specific application client. In order to improve the classification accuracy, researchers have been proposing new methods resorting to several machine learning algorithms and using different traffic features [18–20]. Nonetheless, although the fact that *in the dark* approaches generally require less computational resources is usually a motivation for their development, the search for more accurate behavioral classification methods is also increasing their complexity. In fact, Cascarano et al. [21] compared the computational requirements of a behavioral classifier based on Support Vector Machines (SVMs) and of a DPI classifier and they concluded that both classifiers have similar computational costs.

The scope of this thesis is limited to the fields of traffic monitoring and analysis and P2P networking. The research work presented herein is focused on the study of the challenges in traffic classification raised by P2P traffic and on the different approaches for traffic classification. The limitations of DPI-based methods and the search for new classification approaches that do not resort to payload data motivated the development of traffic classification methods that are not affected by payload encryption and are suitable for real-time operation. The methods proposed in this thesis are based on the analysis of the heterogeneity of traffic features, especially the packet length.

2 Problem Definition and Research Objectives

The problem addressed in this thesis is the classification of P2P traffic in real-time, without using data from the payload of the transport-level protocols. Motivated by the impact of P2P traffic in computer networks, the first studies on P2P traffic classification aimed to characterize the traffic load in networks generated by P2P applications, for which they resorted to the transport-level port numbers [22, 23]. At the beginning of this doctoral programme, however, the uselessness of port numbers as a classification approach was already a reality and, therefore, most published studies addressing the classification of P2P traffic used DPI methods [24–26].

Since the DPI limitations were starting to become a problem, with a few P2P applications adopting encryption techniques, several authors had already started to propose behavioral classifiers as a solution to avoid the computational resources required by DPI and to identify generic P2P traffic without being limited to the known protocols [17].

Many of the behavioral methods proposed in the literature are limited to offline use as they either need to process the entire flows [27, 28] or have to analyze the flows more than once to make a classification [29]. Moreover, most studies process several traffic features, which are used by the classification method as discriminators to distinguish different categories of traffic [15, 30]. In many cases, the proposed methods only address a very specific application protocol [12, 27] or are tested with traffic from a small set of applications [17, 31]. Furthermore, the search to improve the accuracy of behavioral traffic classification has led to an increase of the complexity of the proposed classifiers. Several studies have been proposing different complex methods ranging from statistical study [11, 32] to host interaction analysis [13, 33] and machine learning algorithms [34, 35]. Some of them require a training phase for specific target applications so that the classifier can previously learn the characteristics of the traffic [36, 37]. Being designed for specific protocols or applications, such classifiers cannot identify unknown protocols from a target class, e.g., P2P.

The main objective of this thesis is to present a new classification method for P2P traffic, without being limited to any specific P2P protocol. The proposed method should not use data from the payload of the transport-level protocols, so that it can be used to classify traffic from applications that include payload encryption techniques. Moreover, the new classifier should be suitable for classifying the traffic in real-time, being able to produce results during the lifetime of the flows and without having to analyze them more than once. Additionally, the proposed classification method should use a minimal number of traffic features and avoid complex methods whose computational requirements may grow exponentially with the increase of the amount of processed data.

The following intermediate objectives were defined so as to divide and organize the research work required to accomplish the main objective of this thesis:

1. In order to understand the classification solutions, one of the objectives of this thesis is to study the different traffic classification approaches, their advantages and limitations, and the contexts where the use of each of them may achieve the best performance. The classification methods proposed in the literature and the related works are analyzed so as to learn about the traffic classification topic and to know the state of the art. Furthermore, the problems and challenges for network management raised by the traffic generated by P2P applications and its differences to other communication paradigms are also studied.

2. The second intermediate objective is the analysis of different features of traffic samples generated by P2P and *client-server* applications, which is required for the study of the differences between the traffic of both classes of applications. Given the dual role of the P2P hosts, the experimental traffic is captured at its source, instead of at a network aggregation point.
3. Since P2P hosts act more actively in the network when compared to the ones using typical *client-server* applications, one of the intermediate objectives is to propose a new method that is able to identify hosts running P2P applications. To be consistent with the main objective, the method should be suitable for real-time analysis and not use payload data.
4. The traffic from a host can be generated by a combination of P2P and non-P2P applications. Hence, to accomplish the main goal, it is necessary to present a new method capable of classifying individual flows. Nonetheless, a host running a P2P Voice over Internet Protocol (VoIP) session establishes a flow to another peer, while a host running a P2P media streaming or P2P file-sharing application establishes several flows to other peers. Therefore, the last intermediate objective includes a classification mechanism capable of identifying P2P VoIP flows and P2P media streaming and P2P file-sharing flows.

The applicability of the classification of P2P traffic in network management is not limited to traffic blocking or shaping and may be useful for other techniques for the efficient traffic management as, e.g., content caching [3, 38]. Nonetheless, the problem addressed by this thesis is the classification of P2P traffic and, therefore, the actions taken upon the identification of this kind of traffic and the decisions on what is the most advantageous approach to manage its presence in different network scenarios fall out of the scope of this research work.

3 Thesis Statement

This thesis proposes a new approach for the classification of P2P traffic based on behavioral properties of the traffic. Specifically, the thesis statement is:

The distributed nature of the P2P paradigm influences basic properties of the traffic, like the packet lengths, causing the increase of their heterogeneity. The level of heterogeneity of these properties can be measured through entropy and applied on the characterization of the traffic from P2P applications. The entropy analysis of such traffic features may be used for the purpose of P2P traffic classification in real-time, without using data from the packet payload.

To support this thesis statement, the following research approach was adopted.

The problem and research field are studied and the literature on traffic classification is re-

viewed. The main advantages and limitations of each classification approach are analyzed, as well as the solutions proposed by other researchers.

The biggest difference between P2P and *client-server* architectures is the dual-role played by P2P hosts. Hence, the properties of experimental traffic from both types of applications, collected at its source, are analyzed so as to study how that difference is reflected in network data. Capturing the traffic immediately after the machine that is generating it guarantees that the observed properties result from the applications running in the host and are not affected by the aggregation of several traffic sources. The study included the analysis of several traffic characteristics, focusing on the byte count per time unit, the inter-arrival time, and the packet length.

The observations made in the previous traffic analysis enabled the identification of patterns that are distinct in both types of traffic. Those patterns are related with the heterogeneity of the traffic features, especially the packet length, which is mostly caused by the distributed nature of P2P applications. Larger and more varied traffic traces are captured and the entropy is used to measure the level of heterogeneity of the packet lengths from each host. Since the entropy has to be computed for a fixed number of values, a sliding window with a constant size of N packets is implemented to allow the real-time analysis of the traffic. The entropy computation based on the sliding window is used in all the classifiers proposed in this thesis.

In order to demonstrate the feasibility of using the heterogeneity of the packet lengths, measured through entropy, to distinguish P2P and non-P2P traffic, a host-based classifier is proposed. The classifier uses only four rules, based on the entropy of three different perspectives of the packet lengths, to classify traffic from hosts running P2P applications.

The host-based classifier identifies the traffic from hosts running P2P applications. To classify the individual flows, two separate cases are considered: traffic from P2P VoIP applications and from P2P media streaming or P2P file-sharing applications. In the case of P2P VoIP traffic, the observed heterogeneity results from the speech codec used in the session and, thus, is visible in each flow generated by a VoIP session. In P2P media streaming or P2P file-sharing traffic, however, the heterogeneity is caused by the aggregation of multiple flows used to share contents among peers. Therefore, each of this two types of applications is addressed separately.

To implement a P2P VoIP classifier, the entropy of the packet lengths from traffic traces of several VoIP sessions using different Constant Bit Rate (CBR) and Variable Bit Rate (VBR) speech codecs and applications is analyzed and compiled in intervals. These intervals are used to create behavioral signatures to which the classifier resorts to classify the traffic from each codec.

The analysis of the traffic from P2P media streaming or P2P file-sharing applications is made by capturing traffic traces from several P2P and non-P2P applications and computing the entropy of the packet lengths in different intervals of lengths, and separately for incoming and outgoing packets. Additionally, the entropy of the inter-arrival times and of the remote *host*/*port* pairs is also analyzed. The results of the analysis are used to define rules on which a new classifier for P2P and non-P2P flows is based.

4 Main Contributions

This section briefly describes the main scientific contributions resulting from the research work presented in this thesis.

The first contribution of this thesis is a detailed description of the existing approaches for traffic classification in computer networks and a comprehensive analysis and review of the literature on P2P traffic classification. This study is described in chapter 2, which consists of an article accepted for publication in ACM Computing Surveys [39].

The second contribution of this thesis is the study of traffic from several P2P and non-P2P applications, which was captured at its source, immediately after a host machine running a single or a predefined set of applications. The analysis of the traffic aimed to understand the characteristics of the traffic that are inherent to the application and do not result from the aggregation of traffic from several applications and hosts. Several traffic features, including the byte count per time unit, the inter-arrival time, and the packet length, were analyzed in this study and fitted with a few well-known distributions. This study is described in chapter 3, which consists of an article published in ACM Transactions on Multimedia Computing Communications and Applications [40].

The third contribution of this thesis consists of the observation and analysis of the heterogeneity of the packet lengths in the traffic from P2P and non-P2P applications, the presentation of a real-time method to quantify that heterogeneity by resorting to entropy, and the proposition of a new host-based classifier. The method herein proposed and used in the analysis is based on a sliding window with a constant size of N packets, which makes it possible to compute the entropy in real-time for every packet immediately after the N-th packet. This analysis used traffic from hosts running a single P2P or non-P2P application and presented the differences in the heterogeneity of the lengths of the packets generated by several applications and their effect in the entropy value. An earlier version of this study was presented in an article published in the Proceedings of the 27th IEEE International Performance Computing and Communications Conference (IPCCC 2008) [41] and additional findings were also described in a chapter of the

Handbook of Peer-to-Peer Networking [42]. The traffic analysis was improved and extended to include a larger set of applications and a new traffic classifier capable of identifying the traffic from hosts running P2P applications was proposed. The proposed method does not use payload data, it is based on the entropy of the packet lengths generated by a host, and it uses the same sliding window approach that was previously described to allow the real-time analysis. The extended version of the analysis of the packet lengths from traffic generated by P2P and non-P2P applications and the proposed host-based classifier are described in chapter 4, which consists of an article accepted for publication in The Computer Journal [43].

The fourth contribution of this thesis is the analysis of the traffic from VoIP sessions using different applications and speech codecs, with emphasis on the packet lengths and its dependence on the speech codec used in the session, and the presentation of a VoIP traffic classifier capable of identifying the speech codec used in each session. The packet lengths in the experimental data were analyzed and their heterogeneity was measured using entropy. The compilations of the ranges of packet lengths and entropy were used to propose a new traffic classifier that is able to classify the VoIP flows and identify the speech codec used in the session. The classification method is exclusively based on the packet lengths and on the corresponding entropy and, therefore, does not use payload data. Since the entropy is computed using a sliding window with a limited and constant size and the evaluation of the computational resources used by the classifier showed that the resource consumption grows linearly with the amount of data analyzed, the method can be used for real-time operation. The study of the VoIP session traffic and the proposed classifier are described in chapter 5, which consists of the revised version of an article submitted for publication in an international journal [44].

The fifth and last contribution of this thesis is the proposal of a new classification method that classifies the individual P2P and non-P2P flows without using any payload data. The proposed method is based on the analysis of the heterogeneity not only of the packet lengths, but also of the inter-arrival times and of the remote *host*/*port* pairs to which each local *host*/*port* pair communicates. Furthermore, the heterogeneity of traffic features was analyzed for all the packets in each flow, and also separately for incoming and outgoing packets, and for packets from three ranges of lengths. The entropy was used to measure the heterogeneity of the traffic features in each of the considered perspectives, by resorting to a sliding window. In each iteration of the window, the mean of the entropy since the first iteration was computed and analyzed. The results of the mean of the entropy obtained in the traffic analysis were used to define a set of rules used by the classifier to identify the P2P flows. The study of the traffic and the proposed method are described in chapter 6, which consists of an article submitted for publication in an international journal [45].

5 Thesis Organization

This thesis is organized in seven main chapters. With the exception of the first and seventh chapters, which are devoted to the introduction and conclusions and future work, each of the main chapters is formed by an article published in or submitted to an international journal. Since each article includes its own list of references, the references used in the *Introduction* chapter are listed at the end of chapter 1, so as to keep the consistency with the remaining chapters. For the same reason, the long form of an acronym is repeated in its first occurrence in each chapter. Roman numeration is used for the tables from chapters 2 and 3 included in the list of tables, as this is the numeration used in the layout of the corresponding articles. The subjects and organization of the main chapters of this thesis can be summarized as follows.

Chapter 1 describes the context of this thesis, explaining the scope and focus of the research work and presenting the problem addressed by the thesis and the objectives to be accomplished, as well as the thesis statement and the adopted approach for solving the problem. A summary of the main contributions of this thesis is also included, followed by the description of the organization and structure of the thesis.

Chapter 2 introduces the topic of classification of P2P traffic, presenting the motivation and a brief background for traffic measurements and network monitoring, and focusing on the different approaches for traffic classification. The chapter analyzes the advantages and limitations of each approach and reviews the literature on traffic classification, giving a special attention to the studies focused on P2P traffic.

Chapter 3 first provides a theoretical analysis of the published work on the statistical characterization and modeling of different types of traffic generated by a personal computer. The second part of the chapter is focused on the study of experimental traffic captured at its source. The description of the capturing scenarios and of the collected experimental data is provided, followed by the explanation of the fitting distributions used in the chapter and examples of the results obtained for different traffic features using several types of applications. After that, the summary of the analysis and a discussion of results and distribution parameters is provided.

Chapter 4 focus on the characteristics of the packet length feature and its differences in traffic from P2P and non-P2P applications. The experimental data and the applications used in the study are presented, followed by the description of the heterogeneity of the packet lengths observed in P2P and non-P2P traffic and of the strategy used to quantify the heterogeneity by resorting to entropy. The results obtained for the different applications are presented along with the discussion of the main observations. After that, the chapter proposes a classifier to identify the hosts running P2P applications.

Chapter 5 follows the work described in the previous chapters, focusing on individual flows from VoIP sessions. The analysis of the traffic from VoIP sessions using different codecs is presented, including the description of the experimental data, the explanation of the use of entropy to express heterogeneity, and the analysis of the packet length heterogeneity. After that, a classifier capable of classifying the traffic from VoIP sessions and identifying the speech codec was proposed, followed by the performance evaluation. The chapter includes a supplemental section containing additional information to extend the explanation of specific details of the study.

Chapter 6 provides a study of traffic from P2P media streaming and P2P file-sharing applications, describing the experimental datasets and the applications used in the analysis, the use of entropy to measure the heterogeneity not only of the packet lengths, but also of additional traffic features, the level of observation of the analysis, and the properties observed during the traffic features analysis. Afterwards, the chapter proposes a new classifier to classify P2P traffic flows based on the analysis described in its previous sections, followed by the performance evaluation. Further details and explanations are provided in a supplemental section that finishes the chapter.

Chapter 7 presents the most important conclusions and contributions of this thesis and discusses directions for future research work.

References

[1] A. Kind, X. Dimitropoulos, S. Denazis, and B. Claise, "Advanced network monitoring brings life to the awareness plane," *IEEE Commun. Mag.*, vol. 46, no. 10, pp. 140–146, Oct. 2008.

[2] G. Goth, "Traffic management becoming high-priority problem," *IEEE Internet Comput.*, vol. 12, no. 6, pp. 6–8, Nov.–Dec. 2008.

[3] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet service providers fear peer-assisted content distribution?" in *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2005)*, Berkeley, CA, USA, Oct. 2005, pp. 63–76.

[4] J. Seedorf, "Security challenges for peer-to-peer SIP," *IEEE Netw.*, vol. 20, no. 5, pp. 38–45, Sep.–Oct. 2006.

[5] M. E. Johnson, D. McGuire, and N. D. Willey, "The evolution of the peer-to-peer file sharing industry and the security risks for users," in *Proc. 41st Hawaii Int. Conf. System Sciences (HICSS 2008)*, Waikoloa, HI, USA, Jan. 2008, pp. 1–10.

[6] D. Chopra, H. Schulzrinne, E. Marocco, and E. Ivov, "Peer-to-peer overlays for real-time communication: Security issues and solutions," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 4–12, Jan.–Mar. 2009.

[7] G. Lawton, "Is peer-to-peer secure enough for corporate use?" *IEEE Computer*, vol. 37, no. 1, pp. 22–25, Jan. 2004.

[8] E. P. Freire, A. Ziviani, and R. M. Salles, "Detecting VoIP calls hidden in web traffic," *IEEE Trans. Netw. Service Manag.*, vol. 5, no. 4, pp. 204–214, Dec. 2008.

[9] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2005, vol. 3431, pp. 41–54.

[10] P. Ohm, D. C. Sicker, and D. Grunwald, "Legal issues surrounding monitoring during network research," in *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2007)*, San Diego, CA, USA, Oct. 2007, pp. 141–148.

[11] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting," *Elsevier Computer Netw.*, vol. 53, no. 1, pp. 81–97, Jan. 2009.

[12] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: When randomness plays with you," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Oct. 2007.

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

[14] W. H. Turkett, A. V. Karode, and E. W. Fulp, "In-the-dark network traffic classification using support vector machines," in *Proc. 20th National Conf. Innovative Applications of Artificial Intelligence (IAAI '08)*, Chicago, IL, USA, Jul. 2008, pp. 1745–1750.

[15] A. W. Moore, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," Intel Research, Cambridge, UK, Tech. Rep. RR-05-13, Aug. 2005.

[16] J. Hurley, E. Garcia-Palacios, and S. Sezer, "Host-based P2P flow identification and use in real-time," *ACM Trans. Web*, vol. 5, pp. 1–27, May 2011.

[17] F. Constantinou and P. Mavrommatis, "Identifying known and unknown peer-to-peer traffic," in *Proc. IEEE Int. Symp. Network Computing and Applications (NCA '06)*, Cambridge, MA, USA, Jul. 2006, pp. 93–102.

[18] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Elsevier Performance Eval.*, vol. 67, no. 6, pp. 451–467, Jun. 2010.

[19] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti, "Abacus: Accurate behavioral classification of P2P-TV traffic," *Elsevier Computer Netw.*, vol. 55, no. 6, pp. 1394–1411, Apr. 2011.

[20] X. Li, F. Qi, D. Xu, and X. Qiu, "An internet traffic classification method based on semi-supervised support vector machine," in *Proc. IEEE Int. Conf. Communications (ICC 2011)*, Kyoto, Japan, Jun. 2011, pp. 1–6.

[21] N. Cascarano, A. Este, F. Gringoli, F. Risso, and L. Salgarelli, "An experimental evaluation of the computational cost of a DPI traffic classifier," in *Proc. IEEE Global Communications Conf. (GLOBECOM 2009)*, Honolulu, HI, USA, Nov.–Dec. 2009, pp. 1–8.

[22] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen, "P2P, the gorilla in the cable," in *Proc. National Cable & Telecommunications Association (NCTA)*, Chicago, IL, USA, Jun. 2003, pp. 8–11.

[23] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, Apr. 2004.

[24] T. Karagiannis, A. Broido, N. Brownlee, k. claffy, and M. Faloutsos, "Is P2P dying or just hiding?" in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2004)*, vol. 3, Dallas, TX, USA, Nov.–Dec. 2004, pp. 1532–1538.

[25] A. Spognardi, A. Lucarelli, and R. D. Pietro, "A methodology for P2P file-sharing traffic detection," in *Proc. 2nd Int. Workshop Hot Topics in Peer-to-Peer Systems (HOT-P2P '05)*, La Jolla, CA, USA, Jul. 2005, pp. 52–61.

[26] L. Bin, L. Zhi-Tang, and T. Hao, "A methodology for P2P traffic measurement using application signature work-in-progress," in *Proc. 2nd Int. Conf. Scalable Information Systems (InfoScale '07)*, Suzhou, China, Jun. 2007, pp. 1–2.

[27] A. Bianco, G. Mardente, M. Mellia, M. Munafò, and L. Muscariello, "Web user-session inference by means of clustering techniques," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 405–416, Apr. 2009.

[28] B. Li, M. Ma, and Z. Jin, "A VoIP traffic identification scheme based on host and flow behavior analysis," *J. Netw. Syst. Manag.*, vol. 19, no. 1, pp. 111–129, Mar. 2011.

[29] T. Karagiannis, A. B. M. Faloutsos, and K. claffy, "Transport layer identification of P2P traffic," in *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2004)*, Taormina, Sicily, Italy, Oct. 2004, pp. 121–134.

[30] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 50–60, Jun. 2005.

[31] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying SSH and Skype," in *Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA 2009)*, Ottawa, Canada, Jul. 2009, pp. 1–8.

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

[33] M. Iliofotou, H. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, "Graption: A graph-based P2P traffic classification framework for the internet backbone," *Elsevier Computer Netw.*, vol. 55, no. 8, pp. 1909–1920, Jun. 2011.

[34] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Elsevier Computer Netw.*, vol. 53, no. 14, pp. 2476–2490, Sep. 2009.

[35] M. Mohammadi, B. Raahemi, A. Akbari, H. Moeinzadeh, and B. Nasersharif, "Genetic-based minimum classification error mapping for accurate identifying peer-to-peer applications in the internet traffic," *Elsevier Expert Syst. Appl.*, vol. 38, no. 6, pp. 6417–6423, Jun. 2011.

[36] L. Bernaille, R. Teixeira, I. Akodjenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, Apr. 2006.

[37] M. Dusi, A. Este, F. Gringoli, and L. Salgarelli, "Coarse classification of internet traffic aggregates," in *Proc. IEEE Int. Conf. Communications (ICC 2010)*, Cape Town, South Africa, May 2010, pp. 1–6.

[38] K. Xu, J. Liu, and H. Wang, "Tod-cache: Peer-to-peer traffic management and optimization using combined caching and redirection," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2008)*, New Orleans, LA, USA, Nov.–Dec. 2008, pp. 1–5.

[39] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Detection and classification of peer-to-peer traffic: A survey," *ACM Computing Surveys*, accepted for publication.

[40] J. V. P. Gomes, P. R. M. Inácio, B. Lakic, M. M. Freire, H. J. A. da Silva, and P. P. Monteiro, "Source traffic analysis," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 3, pp. 1–23, Aug. 2010.

[41] J. V. P. Gomes, P. R. M. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "Analysis of peer-to-peer traffic using a behavioural method based on entropy," in *Proc. 27th IEEE Int. Performance Computing and Communications Conf. (IPCCC 2008)*, Austin, TX, USA, Dec. 2008, pp. 201–208.

[42] J. V. P. Gomes, P. R. M. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "The nature of peer-to-peer traffic," in *Handbook of Peer-to-Peer Networking*, X. Shen, H. Yu, J. Buford, and M. Akon, Eds. Berlin Heidelberg: Springer, 2010, pp. 1231–1252.

[43] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Exploring behavioral patterns through entropy in multimedia peer-to-peer traffic," *The Computer J.*, accepted for publication.

[44] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Identification of peer-to-peer VoIP sessions using entropy and codec properties," submitted for publication.

[45] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Classification of one-to-many peer-to-peer traffic using packet length and entropy," submitted for publication.

Chapter 2

Detection and Classification of Peer-to-Peer Traffic: A Survey

This chapter consists of the following article:

Detection and Classification of Peer-to-Peer Traffic: A Survey

João V. Gomes, Pedro. R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

ACM Computing Surveys, accepted for publication, 2012.

According to 2010 Journal Citation Reports published by Thomson Reuters in 2011, this journal scored ISI journal performance metrics as follows:

ISI Impact Factor (2010): 8.000

ISI Article Influence Score (2010): 4.366

Journal Ranking (2010): 1/97 (Computer Science, Theory & Methods)

A

Detection and Classification of Peer-to-Peer Traffic: A Survey

JOÃO V. GOMES, PEDRO R. M. INÁCIO, MANUELA PEREIRA and MÁRIO M. FREIRE,
University of Beira Interior and Instituto de Telecomunicações
PAULO P. MONTEIRO, Nokia Siemens Networks, University of Aveiro and Instituto de Telecomunicações

The emergence of new Internet paradigms has changed the common properties of the network data, increasing the bandwidth consumption and balancing traffic in both directions. These facts raised important challenges, making it necessary to devise effective solutions for managing network traffic. Since the traditional methods are rather ineffective and easily bypassed, particular attention has been paid to the development of new approaches for traffic classification. This article surveys the studies on peer-to-peer traffic detection and classification, making an extended review of the literature. Furthermore, it provides a comprehensive analysis of the concepts and strategies for network monitoring.

Categories and Subject Descriptors: A.1 [**Introductory and Survey**]; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*; *Packet-switching networks*; C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management*; *Network monitoring*; C.4 [**Performance of Systems**]: *Measurement techniques*

General Terms: Management, Measurement, Security

Additional Key Words and Phrases: Application classification, deep packet inspection, behavioral analysis, peer-to-peer, traffic monitoring

ACM Reference Format:
Gomes, J. V., Inácio, P. R. M., Pereira, M., Freire, M. M., and Monteiro, P. P. 2011. Detection and classification of peer-to-peer traffic: A survey. ACM Comput. Surv. V, N, Article A (January YYYY), 39 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

In the early years of Internet, network connections relied on the *client-server* paradigm, generating an asymmetric amount of data in both upstream and downstream directions. Nonetheless, users became more influent, not only on the information available on Internet, but also on its distribution. The so-called *Web 2.0* offered Internet hosts the opportunity to provide their own multimedia contents and to directly interact with other peers. Furthermore, the popularity gained by peer-to-peer (P2P) systems in the end of the last century enabled the direct distribution and sharing of

This work was partially supported by *Instituto de Telecomunicações*, by University of Beira Interior, and by *Fundação para a Ciência e a Tecnologia*, through the grant contract SFRH/BD/60654/2009 and the project TRAMANET: Traffic and Trust Management in Peer-to-Peer Networks with contracts PTDC/EIA/73072/2006 and FCOMP-01-0124-FEDER-007253.
Authors' addresses: J. Gomes (jgomes@penhas.di.ubi.pt), P. Inácio (inacio@di.ubi.pt), M. Pereira (mpereira@di.ubi.pt) and M. Freire (mario@di.ubi.pt), *Instituto de Telecomunicações*, Department of Computer Science, University of Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal; P. Monteiro (paulo.1.monteiro@nsn.com), Nokia Siemens Networks Portugal, S. A., Rua Irmãos Siemens, 2720-093 Amadora, Portugal.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© YYYY ACM 0360-0300/YYYY/01-ARTA $10.00
DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

contents between Internet users. The once passive user has gained a new and very active role in the Internet, acting simultaneously as client and server. These important changes in the services running over the Internet and in the behavior of the end-hosts modified the traditional properties of network traffic, which is evolving towards a more balanced bandwidth usage in both directions. Additionally, most of these applications present a greedy profile, consuming as much bandwidth as they can, which may end up interfering with priority policies. Azzouna and Guillemin [2003], for example, found that 49% of the traffic in an Asymmetric Digital Subscriber Line (ADSL) link was caused by P2P applications, while Gerber et al. [2003] and Sen and Wang [2004] observed the growth and prevalence of this kind of traffic. In 2007, *ipoque* conducted a world wide study about the Internet traffic [Schulze and Mochalski 2007] and the results showed that P2P file-sharing applications were producing more traffic than all the other applications together, being responsible for 49% to 83%, on average, of all Internet traffic, and reaching peaks of over 95%. Another study by *ipoque* [Schulze and Mochalski 2009], in 2008 and 2009, concluded that, although the total amount of traffic generated by P2P file-sharing has increased, its percentage has decreased to an average value of between 42.51% and 69.95%. This fact may be due to an increase of the traffic generated by video streaming and file hosting web services, like *YouTube*, *Tudou*, or *RapidShare*. Yet, there have been several discussions regarding the adoption of P2P solutions by some of the these services, namely *YouTube* and *Tudou*, in order to accelerate their downloading rates and reduce the transmission cost. In fact, the web-based CNN live channel service relies now on the P2P paradigm due to a plug-in each user has to install.

In spite of the share of global traffic of each Internet application, P2P systems motivate particular attention from the perspective of network management for the dual role their peers play. For a certain amount of data downloaded by a peer, a portion of data is also uploaded by the same peer. Instead of being concentrated in a dedicated server, the distribution cost of the service is thus shared by the users. While this fact is advantageous for content providers, it implies that a host receiving a service will produce additional traffic in its Internet Service Provider (ISP) network or Local Area Network (LAN) as it is also providing the service to a different peer. Moreover, hosts in P2P networks usually receive and provide contents from and to several peers at the same time. Hence, P2P applications are likely to produce a much larger number of connections than typical *client-server* applications. In addition, mechanisms to search contents in remote peers also cause an increment of the communications between hosts. These facts make P2P traffic management more challenging than traffic from *client-server* applications, which is usually formed by a single or a few connections. Besides of the increase of the bandwidth consumption, the amount of traffic generated by P2P applications in both directions is more balanced, as opposed to the greater weight in downstream of the traditional *client-server* traffic. This difference poses an important issue in terms of traffic management, as most networks (or Internet connections) were devised to offer lower bandwidth in upstream. Managing the network and implementing specific policies for P2P traffic does not necessarily means it should be blocked or heavily throttled. Nevertheless, there are techniques that can help to efficiently manage this traffic if one is able to classify it, as content caching [Karagiannis et al. 2005b; Xu et al. 2008].

Although the traffic management issues are of particular concern mainly for ISPs and network administrators [Karagiannis et al. 2005b; Freire et al. 2009], there are other problems, mostly related to security risks and vulnerabilities [Zhou et al. 2005; Seedorf 2006; Li et al. 2007; Johnson et al. 2008; 2009; Chopra et al. 2009], that are magnified by the distributed nature of P2P systems and by the role of their peers, and that may affect companies and home users. While reducing the overlay distances

between end hosts for the exchange of contents, the P2P paradigm also amplifies the effects of virus and other threats by facilitating their dissemination. Ensuring privacy, anonymity or confidentiality is also more difficult in these networks and constitutes a real concern, not only for home users, but also for companies [Lawton 2004]. These problems do not result directly from the P2P communication paradigm, but they are a consequence of the proximity between peers and of the simplicity of content sharing in P2P systems. This fact, together with the multiple connections created by P2P applications and the encryption and obfuscation techniques used by most of them, make it more difficult to identify threats in the traffic.

In this context, traffic classification based on the application protocol appears as a crucial tool to manage the data within the networks, to fairly share the available bandwidth, to assure the Quality of Service (QoS), to implement billing mechanisms or to deploy security measures. However, identifying the application that generated the traffic is nowadays a difficult task and may have several associated issues (e.g., random port number or payload encryption) as described by Kind et al. [2008]. The traditional and most obvious method to classify network traffic was to associate the transport port numbers to well-known application protocols. However, this approach became ineffective as soon as a significant number of applications started to use random port numbers, or port numbers used by other well-known protocols. Karagiannis et al. [2004a] identified P2P applications running on port 80 and estimated that 30% to 70% of the overall P2P traffic is generated by applications using random port numbers. Likewise, the results by Madhukar and Williamson [2006] show that the same percentage of Internet traffic cannot be correctly identified by port based methods. More recently, Basher et al. [2008] concluded that 90% of the P2P traffic may be using random ports.

Therefore, in the last years, the classification of Internet Protocol (IP) traffic has been a very active research field, with many contributions based on distinct approaches. When port-based mechanisms lost their effectiveness, the solution was to employ Deep Packet Inspection (DPI) techniques, which were frequently used by Network Intrusion Detection Systems (NIDSs) for security purposes, to identify the traffic using signatures in the contents of the packets. However, this approach also has a few important drawbacks, mainly related with the computational resources required to inspect traffic in high-speed networks, with the impossibility to accomplish their purpose when the payload is encrypted and with privacy issues. The alternative was to design different statistical or behavioral (based on heuristics) methods, which resort to the packet header and flow-level data to segregate the traffic into different classes.

The main contribution of this article is to survey the existing studies, methods, techniques and applications on the topic of traffic classification. Although several concepts and techniques may also apply to other fields of traffic monitoring, herein they will be analyzed from the perspective of traffic classification. Most of the classification methods may be applied to the classification of the traffic from different types of applications. Nonetheless, since P2P systems are on the basis of a large number of research contributions, a special attention will be given to the studies addressing the subject of P2P traffic classification.

In order to facilitate the understanding of the survey, it is included an introduction to the subject of network measurement from the perspective of traffic monitoring (and, more specifically, classification), which explains a few important concepts and techniques. The existing approaches for traffic classification are also carefully described in the survey, explaining their way of functioning, in which situations they are more valuable and what are their limitations. Foremost, this article provides an extended review of the literature, presenting the available methods and their performance, and organizing them based on the type of analysis they perform.

The remainder of the article is structured as follows. Section 2 describes the related work. Section 3 gives an explanation of important concepts and techniques for traffic measuring, while section 4 describes the distinct approaches for traffic classification together with their main advantages and weaknesses. An analysis of the published literature is presented in section 5, followed by the Conclusions section.

2. RELATED WORK

The topic of traffic classification has aroused considerable interest in recent scientific contributions, with several studies addressing the challenges raised by new application protocols and proposing novel techniques and solutions for its classification. Nonetheless, there are still few papers surveying the existing works on the field, as well as analyzing distinct methods and approaches.

The Internet Measurement Research Group (IMRG) of the Internet Research Task Force (IRTF) sponsored a workshop on *Application Classification and Identification*. The report of this workshop [Strayer et al. 2008] described a number of important topics, highlighting the challenges inherent to the task of traffic classification and its main motivations and summarized the contribution of each paper.

Madhukar and Williamson [2006] compared the efficiency of three distinct techniques for the identification of P2P traffic: port numbers, payload signatures, and transport-layer heuristics. In order to provide a longitudinal study of the performance of each technique, they collected traffic traces during two years and used them as sample data to evaluate each method. Kim et al. [2007; 2008] also performed a comparative study between three different approaches to traffic classification: port-based, behavioral, and statistical. The evaluation was based on available applications and research tools and techniques: *CoralReef* [Moore et al. 2001], *BLINC* [Karagiannis et al. 2005a], and Machine Learning (ML). The authors tested the solutions using seven distinct traffic traces from two backbone and two edge links from United States of America, Japan, and Korea. In [Li et al. 2009], four different classification methods were also compared in terms of efficiency and effectiveness: well known port numbers, DPI, Naïve Bayes and the *C4.5* decision tree method. In order to evaluate the performance of the mechanisms from both temporal and spatial perspectives, the authors used traffic traces collected over several years on two different sites.

A survey on traffic classification solutions relying on ML was provided in [Nguyen and Armitage 2008b]. Although the study was especially focused on the identification of application-level protocols through the use of ML techniques, the authors also included a description of the difficulties imposed by many recent Internet applications and the main reasons for developing new methods for the classification of the traffic generated by those applications. Cascarano et al. [2010b] compared the performance of three different traffic classifiers for peer-to-peer television (P2PTV) applications: a DPI mechanism, a method based on single-class Support Vector Machines (SVMs), and a method based on multi-class SVMs. They evaluated three P2PTV applications and used traffic traces collected at the border gateway of a LAN of a university campus.

The most closest work to the study presented herein was the one by Callado et al. [2009]. After introducing the subject of traffic analysis, the authors described the state-of-the-art of flow-based traffic analysis, pointing out several flow properties of Internet traffic. They also described many research works on the traffic classification field and provided a theoretical comparison of the results obtained by four distinct studies.

This survey distinguishes itself from the previous works for its wide and comprehensive analysis, and for giving special attention to the identification of P2P traffic and its challenges. Moreover, as traffic classification is a very active research topic many works described herein are subsequent to [Callado et al. 2009]. Unlike most studies, this survey starts by introducing the subject of traffic measurement from the perspec-

tive of traffic classification, so that basic concepts, important for the correct reading of the remainder of the paper, can be well understood. Besides describing the existing approaches for traffic classification and identifying its main advantages and weak points, this survey provides a broad review of the literature. Furthermore, it analyzes, compares and gives a structured view of studies, approaches, techniques and available applications for the classification of P2P network data.

3. MEASURING FOR NETWORK MONITORING

Solid research studies on the characteristics and behavior of computer networks, as well as the development of effective mechanisms for the traffic management and the design of better and more efficient networks, require strong and accurate traffic analyses and collections. Over the last few decades, many authors addressed the subject of network (and, more specifically, Internet) traffic measurements, highlighting its crucial role for understanding the behavior of computer networks, e.g., [Jain and Routhier 1986; Claffy and McCreary 1999; Cáceres et al. 2000; Williamson 2001; McGregor 2002; Paxson 2004].

However, measuring network traffic is far from being a simple problem. Corroborating this idea, Paxson [2004] describes a few challenges one has to deal with when performing such task, as well as some interesting strategies for a sound Internet measurement. McGregor [2002] also discusses several technical issues, while proposing guidelines for quality measurements.

Likewise, also in the context of traffic classification, and in spite of playing an essential role in a solid work, network measurements can be a source of technical challenges [Arlitt and Williamson 2007]. In the next subsections, the topic of traffic measurement is explored from the point of view of traffic classification, considering important concepts, techniques, and approaches. Nevertheless, for a deeper discussion on the subject of network measurement, we refer to the book by Crovella and Krishnamurthy [2006], as well as to the references cited in this section.

3.1. Traffic Measurements

At this point, it is useful to distinguish between different approaches for network traffic measurement or monitoring. Based on a few specific characteristics, Williamson [2001] classifies the research tools for network study into the following categories: *hardware* or *software*; *protocol level*; *LAN* or *Wide Area Network (WAN)*; *on-line* (or real-time) or *off-line*; and *passive* or *active*. The discussion of each of these categories may be appropriate or not, depending on the purpose of each monitoring study or tool. However, in most studies [Claffy and McCreary 1999; Paxson 2004; Duffield 2004; Bartlett et al. 2007b], authors differentiate, mainly, between active and passive measurements. Herein, these aspects will be briefly discussed from the perspective of traffic classification.

3.1.1. Hardware and Software based Solutions. Practitioners and researchers working in the field of traffic classification are more interested in analyzing the IP packets or the Ethernet frames. Hence, it is not significant if the traffic measurements are made using hardware or software based tools.

Nonetheless, dedicated hardware solutions tend to present a better processing performance, which is useful for real-time analyses. A few companies, like *ipoque* [2011], *Endace* [2011], *Napatech* [2011], or *WildPackets* [2011], provide hardware systems for traffic monitoring or high-speed network interfaces with dedicated buffers for traffic capturing, like the Data Acquisition and Generation (DAG) cards. In terms of traffic classification, a few authors also resort to hardware devices, like Field Programmable

Gate Arrays (FPGAs) or Ternary Content Addressable Memory (TCAM), to improve the computational performance of DPI mechanisms [Yu 2006; Mu et al. 2007].

3.1.2. Protocol Level. It is possible to measure the traffic at different, and even multiple, protocol levels. However, since traffic classification is mostly used for Internet traffic, measurements for that purpose are usually made at the Ethernet or IP levels.

3.1.3. LAN and WAN. For the purpose of traffic classification, measurements can be conducted, with no lost of information or research knowledge, in LANs instead of WANs, which typically are not so easy to get access to.

3.1.4. On-line and Off-line Analyses. Although, in terms of the traffic measurement, on-line and off-line approaches do not differ significantly, the latter is more used whenever a real-time analysis is not necessary, since such task would require higher computational power to be accomplished in high-speed links. Moreover, the usage of off-line trace files is crucial for research and validation purposes, as it allows one to run different analyses through the same data and compare the obtained results.

Nevertheless, on-line measurements are obviously imperative for, e.g., NIDSs, firewalls, or other devices responsible for traffic management, which need to take immediate actions (e.g., drop or forward packets) on the network traffic. However, in these cases, the use of on-line measurements may impact the performance of high-speed networks.

3.1.5. Active and Passive Measurements. The active approach resorts to the injection of actual packets into the network, in order to observe the behavior of the network, hosts or applications. This kind of measurements is mainly used for monitoring the performance of the network or to identify weak points in the system, being especially suitable for the evaluation of QoS levels. `ping` and `traceroute` are simple examples of tools that implement active measurements.

Since active methods rely on the use of artificial traffic, they allow one to easily control the simulation of the scenarios that he or she wants to analyze or to test, like the traffic class, nature, frequency, etc. However, such traffic will not directly reflect the behavior nor the influence of the application and of the human behavior. Moreover, these methods will increase the traffic load in the network, which may affect not only the available bandwidth, but also the performance of routers, switches, or other network equipment. In the case of large networks, administrators can face scalability problems when using active measurement techniques.

Passive measurement techniques do not produce any additional traffic. Instead of injecting packets into the network, a passive monitor simply looks at the traffic and collects data that can be used to infer on the behavior of hosts, applications, network performance or even on the user influence in the generated traffic. It does not send additional data to the network being monitored, modify any contents, interfere in the packets route (unless it has also other functions, as firewall, gateway, etc.), or increase the traffic load. Furthermore, an important advantage of this kind of approach is that the final data reflects the properties of the real traffic. Passive measurements are, therefore, particularly useful for traffic management, retrieving important knowledge about the behavior of the traffic.

Nevertheless, passive measurements may produce large amounts of data, which may require ambitious computational resources not only to store and handle that data, but also to process it and generate useful conclusions. For the same reason, its analysis in real-time may be a demanding task. In some contexts and for some purposes, the usage of real traffic may also raise a few legal issues [Ohm et al. 2007].

3.2. Per Packet and Per Flow Analysis

Measurements made for the purpose of Internet traffic analysis are mainly focused on IP packets or Ethernet frames. The traffic under analysis is usually captured and stored on a packet-by-packet manner, as the most obvious method to accomplish the task of capturing traffic is to simply catch each individually data unit traveling in the network. Some of the existent tools for network management include means to display, process, statistically analyze, or even make decisions on each packet individually. This *per packet* approach is especially interesting for applications like NIDSs (e.g., *Snort* [2010] or *Bro* [2010]), which need to process and decide upon each packet. Also, sniffers or protocol analyzers especially designed for off-line analysis, like *Wireshark* [2010] or *Ettercap* [2010], usually inspect each packet deeply, gathering information from all the layers of the protocol stack.

Although packets are individual data units when traveling through the network, a relation exists between many of them [Jain and Routhier 1986]. Usually, they are generated by the same request or application, they contain acknowledgement messages from reliability mechanisms (like it happens with Transmission Control Protocol (TCP) traffic), or they are simply carrying an amount of data that is too large to fit in a single Ethernet frame. Therefore, the relation between the packets comprises a relatively hidden knowledge about the network and the traffic behavior, which can be assessed by analyzing the traffic in terms of data flows.

A flow is, most of the times, defined as a set of packets that share a common key: source and destination IP addresses and transport port numbers [Claffy and McCreary 1999; Duffield 2004; Duffield et al. 2005; IETF 2008]. It is considered *active* while the time interval between each packet belonging to the flow is lower than a certain threshold. The timeout value may depend on the purpose of the analysis. Although a few studies propose distinct timeouts, Claffy et al. [1995] explored different values and identified 64 seconds as a good compromise between the size of the flow and the effort to initialize and terminate the flows. Furthermore, a flow may also be defined as *unidirectional* or *bidirectional*, depending on whether one wants to consider the packets traveling between two *address-port* pairs in each direction as two independent flows, or the packets in both directions as a single flow [Apisdorf et al. 1996; Claffy et al. 1995]. Because of the usual asymmetry of the traffic exchanged between two addresses in typical *client-server* connections and also due to the asymmetric routes in the core Internet, unidirectional flows are mostly used in studies on network performance and bandwidth management, for which it is useful to measure the differences in the traffic in both directions [Claffy et al. 1995]. On the other hand, bidirectional flows are a natural option to represent TCP sessions and, for the purpose of traffic classification, they are a more logical approach to follow, as the traffic exchanged between two *address-port* pairs, in both directions, belongs to the same traffic class and was generated by the same application. Nonetheless, Smith et al. [2001] were able to successfully use unidirectional packet headers traces to analyze TCP transactions.

In order to analyze the traffic from a flow perspective, a monitoring tool can still capture the packets individually, but it has to organize them in a table of flows, based on the source and destination information (address and port). Several tools, e.g., CoralReef [Moore et al. 2001], were developed to perform flow-based analyses of traffic from network adapters or from off-line packet traces. However, it is possible to receive the flow information directly from routers or other network elements, e.g., using a flow export protocol, like *Cisco NetFlow* [2010] or the Internet Protocol Flow Information eXport (IPFIX) [IETF 2008], a standard for exporting flow data currently under development. *NetFlow* data can be read and analyzed by a few existent applications, like *Flow-tools* [Romig et al. 2000] or *FlowScan* [Plonka 2000].

3.3. Collecting Traffic Data

The access to the network data for traffic measuring, as mentioned in a few studies [Duffield 2004; McGregor 2002], may be performed by copying the transmission signal (e.g., through the use of a splitter) and analyzing it on a dedicated network monitor, by using a router or a switch to copy all the traffic to an output interface, or by directly tapping a shared link. Nevertheless, there are also a few global infrastructures for the active measuring of Internet, that collect data from world wide links [Murray and Claffy 2001]. The datasets containing traffic from computer networks should be carefully handled in order to protect the privacy of the users, as well as other sensitive data. Several considerations and good practices regarding this subject are discussed in [Allman and Paxson 2007].

As seen in previous subsections, the passive data collection can be made by polling routers to obtain flows data, or by packet capturing. While in the former approach, data is usually acquired through the use of protocols like IPFIX, in the latter, the trace files are collected using commercial or public domain network traffic capturing software, like *tcpdump* [2011] and its Windows version, *WinDump* [2011], or even other available tools developed with basis on the *libpcap* [tcpdump 2011] or *WinPcap* [2011] libraries.

Although the most natural means is to capture the complete packet, such technique generates large trace files, which would require larger storage capacity and processing power to handle the traffic in high-speed links. Moreover, the increasing integration of measurement techniques into routers, switches and other network elements that do not possess a high processing power [Duffield 2004; Jurga and Hulbój 2007] motivates the development of solutions that can reduce the amount of data collected, as described in the next subsection.

3.4. Trace Reduction

The most common approaches for trace reduction resort to packet filtering or to the minimization of the data that is kept for future analysis [Duffield 2004; Arlitt and Williamson 2007]. It is possible, depending on the specific goals of each study, to monitor exclusively the packets from a given application. However, such selection is usually made using the transport layer port numbers, which is consensually considered a naive approach. Alternatively, one may select only the packets that establish or finalize a connection or a request, or use any other selection criterion that may be more coherent with the objective of a particular analysis and decrease the number of packets to be captured.

The amount of data stored can be reduced by saving a summary of each application protocol-specific request; by capturing a limited portion of the packet or even only the headers of the first layers of the TCP/IP protocol stack; or by keeping information of a flow instead of storing each packet that belongs to it.

A particular case of packet filtering is the use of packet sampling methods [Amer and Cassel 1989], whose objective is to randomly (or pseudo-randomly) choose a small set of the packets observed in the measuring point. It is intended that the set of packets obtained be as much representative as possible of the traffic one plans to measure. There are different packet sampling techniques which may be more useful in distinct cases, depending on factors like the goal of the study, the network state, the traffic characteristics or the resources constrains. Jurga and Hulbój [2007] elaborated on the existent methods for packet sampling and their application in network measurements. Duffield [2004] addressed the subject of Internet traffic sampling as well, providing a long and sound structured discussion of several important topics on passive traffic measurement.

Table I. Side-by-side comparison of the approaches for traffic classification.

Approaches	Characteristics	Advantages	Weaknesses
Port number matching	— associates port numbers with applications	— low computational requirements — easy to implement	— lack of classification performance due to random port numbers
Deep packet inspection	— relies on payload data	— high classification performance	— may not work for encrypted traffic — requires high processing resources — can only be used for known applications
Classification *in the dark*	— uses only packet header and flow level information	— usually lighter than DPI — applicable for encrypted traffic — can identify unknown applications from target classes	— usually has lower classification performance when compared to DPI
Active crawlers	— based on modified instances of the target applications	— identifies accurately users of the target applications	— identifies only the traffic exchanged with the crawler — injects additional traffic in the network

4. TRAFFIC ANALYSIS AND CLASSIFICATION APPROACHES

In the early times of the Internet, traffic classification was a straightforward task that was easily accomplished by matching the port numbers of the transport protocols with the application protocols. However, since many Internet applications, especially the ones based on the P2P architecture, evolved to use random port numbers or ports assigned to well known protocols (e.g., Hypertext Transfer Protocol (HTTP)), identification strategies agnostic to the port numbers became more common. The most natural approach is to look inside the packets and see what type of data they carry and which application protocol was used. Regardless of that, several statistical or behavior-based methods that do not inspect the contents of the packets have been developed more recently. Table I provides a simple side-by-side overview of the main characteristics of each classification approach. For a better understanding of the remaining of the paper, a discussion on the different types of techniques for traffic classification, the way they operate, their advantages and their drawbacks is provided in the following subsections. Furthermore, two additional subsections were included to address the topic of ground truth verification and describe the most common metrics for the evaluation of the performance of a classification mechanism.

4.1. Traffic Classification Based on Port Numbers

The classification of network traffic based on the User Datagram Protocol (UDP) or TCP port numbers is a simple approach built upon the assumption that each application protocol uses always the same specific transport layer port. This method was mostly useful in the identification of well known protocols like, e.g, HTTP or Simple Mail Transfer Protocol (SMTP), which use the port numbers 80 and 25, respectively. However, many Internet applications easily bypass this identification strategy by simply using random or unknown port numbers, disguising their traffic using port numbers generally used by other well known protocols (e.g., port 80) that are usually allowed by firewalls. Thereby, port numbers as a classification mechanism are consid-

```
alert udp $HOME_NET any -> $EXTERNAL_NET any (msg:"LocalRule:P2P eDonkey UDP
outbound - Status Request"; flow:to_server; content:"|E3 96|"; depth:2;
classtype:policy-violation; sid:1000019; rev:1;)
```

Fig. 1. Example of a *SNORT* rule to detect a payload signature for the traffic generated by *eDonkey* with obfuscation, proposed in [Freire et al. 2009].

ered obsolete [Karagiannis et al. 2004b; Moore and Papagiannaki 2005; Madhukar and Williamson 2006].

4.2. Traffic Classification Based on Deep Packet Inspection

DPI methods, usually the most accurate, are based on the inspection of the packets payload. They rely on a database of previously known signatures that are associated to application protocols, and search each packet for strings that match any of the signatures. This approach is used not only in the classification of network traffic, but also in the identification of threats, malicious data and other anomalies. Because of their effectiveness, classification systems based on DPI are especially significant for accounting solutions, charging mechanisms, or other purposes for which the accuracy is crucial. Fig. 1 shows an example of a *SNORT* rule for the detection of a data signature in the traffic from *eDonkey* with obfuscation mechanisms enabled.

However, deeply inspecting each packet can be a demanding task in terms of computation power and may be unfeasible in high-speed networks. Therefore, some mechanisms search only a part of each packet or only a few packets of each flow as a compromise between efficiency and accuracy. Besides of the performance issues, the inspection of contents of the packet may also raise legal issues related with privacy protection [Ohm et al. 2007].

Nevertheless, the main drawback of DPI techniques is their inability to be used when the traffic is encrypted. Since, in these cases, the contents of the packets are inaccessible (encrypted), DPI-based mechanisms are restricted to specific packets of the connection (e.g., when the session is established) or to the cases when UDP and TCP connections are used concurrently and only the TCP sessions are encrypted. Packets with no payload, which may be malicious, cannot be classified as well. DPI methods are also sensitive to modifications in the protocol or to evolution of the application version: any changes in the signatures known by the classifier will most certainly prevent it from identifying the application. Moreover, DPI methods that rely on signatures for specific applications, can only identify traffic generated by those applications.

4.3. Traffic Classification in the Dark

The inspection of the contents of IP packets, as discussed in the previous subsection, is not always a valid option for the identification of application-level protocols. Therefore, new methods that do not resort to the deep inspection of the packets have been developed. The strategy of this kind of approach, sometimes called *in the dark* [Karagiannis et al. 2005a; Turkett et al. 2008], is to classify the traffic using behavioral or statistical patterns, based on flow-level data or generic properties of the packets [Moore et al. 2005], like addresses, ports, packet size, etc.

The main advantage of classification *in the dark* is the ability to identify a protocol without the need to examine the contents of the packet. As a consequence, mechanisms based on this approach cannot aspire to the same accuracy level of DPI methods. Their results should be understood as a strong suspicion regarding the probable application protocol. Nevertheless, recent studies have achieved high success rates in the classification of Internet traffic. Additionally, classification *in the dark* can more easily be applied to unknown applications since many methods based on this approach classify

the traffic in classes of applications (e.g., web traffic, email, video streaming, P2P, etc.) instead of specific applications.

The existent mechanisms use distinct techniques to correlate the traffic properties and conclude on the application protocol, such as statistical measures, sets of heuristics, or machine learning algorithms. The following subsections introduce each of these approaches.

4.3.1. Statistical Mechanisms. Statistical methods usually rely on flow and packet level properties of the traffic, like flow duration and size, inter-arrival times, IP addresses, TCP and UDP port numbers, TCP flags, packet size, etc. These properties are used, individually or combined, to calculate statistical values, from simple measures as average or variance, to more complex ones like the probability density function. In some studies [Crotti et al. 2006], statistical models of the traffic from a certain application are created. Generally, such approach requires a learning phase to build a reference model that can be used to classify unknown traffic.

4.3.2. Heuristics Based Methods. Many behavioral mechanisms for traffic classification are based on a predefined set of heuristics. Although a large part of them are common to the majority of the research works, distinct combinations or sets are proposed in several studies. Typical heuristics include the network diameter, the presence of nodes acting both as client and server, the number of hosts a user communicates with, the source-destination IP pairs that use both TCP and UDP, the number of distinct addresses and ports a user is connected to, etc. Generally, the set of heuristics is checked sequentially, and, depending on the result, the packet (or flow) is classified as belonging, or not, to a certain application-level protocol.

4.3.3. Machine Learning Techniques. A large part of the studies propose classification mechanisms based on different supervised or unsupervised ML techniques, such as Bayesian estimators or networks [Moore and Zuev 2005; Auld et al. 2007], clustering [McGregor et al. 2004], decision trees [Branch et al. 2009], etc. They assemble a set of traffic characteristics which they correlate using of probabilistic functions, associating each packet or flow to a certain class.

4.4. Traffic Classification Using Active Crawlers

The majority of the solutions in the literature are passive, as they do not interfere with the data within the network neither they generate any additional traffic. Nevertheless, some authors have also developed active mechanisms that crawl the network to collect data used to classify the traffic [Saroiu et al. 2003]. A few of them implemented fake or modified instances of the target applications whose main purpose is to identify hosts running the original applications [Ohzahata et al. 2005].

This kind of approach is generally used for very specific purposes, such as the identification of users running a certain application. Some authors resorted to active crawlers to collect statistics on the number of hosts running the target application and on the properties of the connections to peers (available bandwidth, latency, etc.) [Saroiu et al. 2003].

4.5. Ground Truth Verification

The use of pre-collected traffic from computer networks is of critical importance for the creation and testing of new methods for traffic classification in respect to the application-level protocols. Nonetheless, without the ability to assess its ground truth application information, the use of traffic data is of limited value [Sperotto et al. 2009].

The majority of the packet traces publicly available are limited to the headers due to privacy concerns, making it difficult to obtain the associated ground truth regard-

ing the application. For that reason, in most studies on traffic classification, the researchers collect their own traffic traces to test the accuracy of their solutions. Such approach makes the comparison of different methodologies inconsistent as the performance of each of them was evaluated in different conditions [Salgarelli et al. 2007]. The use of methods to accurately verify and label the ground truth information of packet traces before making the headers publicly available, would solve the problem while still keeping the private data.

In many studies, the ground truth verification is obtained by using an alternative method as reference baseline, e.g., port number matching or DPI [Karagiannis et al. 2005a]. However, such approach will depend on the accuracy of the classifiers used as baseline. Port number matching, e.g., is now considered an ineffective option, while DPI may be unsuitable for encrypted traffic. In fact, when a novel mechanism for traffic classification is proposed, under the pretext that the existent solutions are not completely effective, it is nonsensical to test the accuracy of the new method by using an existing one as the baseline for performance comparison.

Alternatively, hand-classification may be used to verify the ground truth information of the traces [Moore and Zuev 2005]. However, the process can be slow and tiresome. Moreover, it is also possible to create traffic collections from a small network of computers, running a predefined set of applications in a controlled environment. Nonetheless, the obtained traces may not exhibit properties that reflect the human behavior.

Given the increasing concern regarding this topic, a few authors have, more recently, addressed the subject of ground truth verification of application traffic. Canini et al. [2009] presented *GTVS*, a framework to improve and simplify the process of associating traffic data with application-level protocols. It makes use of the packet payload inspection and of multiple heuristic rules to infer the ground truth information and it provides a graphical interface to facilitate manual verification of traffic traces. Gringoli et al. [2009] proposed *GT*, a toolset to assess the ground truth of application traffic. Its architecture differs from *GTVS* mainly in the fact that it includes the existence of a daemon, which is supposed to run in each client and return the information of the process that originated each network connection. Although this approach may significantly increase the accuracy of the verification, the deployment of the client daemon may be difficult in most contexts, or even near impossible in large networks. A similar approach was followed by Szabó et al. [2008], who also described a client-based solution. In this case, the authors suggested the implementation of a client driver that inserts a byte mark in each outgoing packet whose size is not yet the size of the Maximum Transmission Unit (MTU), so that it can avoid the IP packet fragmentation.

All these approaches have their merits and weaknesses, but none is perfect, though. Relying on an alternative classifier to work as baseline reference enables the ground truth identification in every trace, independently of the size of the network. However, if the reference classifier uses DPI, the payload data in the traces must not be encrypted nor removed. Moreover, the evaluation of the performance of the new classification method will always depend on the accuracy of the reference classifier, which may also loose effectiveness when applications evolve and change the properties of their communications (at payload level or even behavioral level). If the results show a certain misclassification rate (even if it is very low), it is impossible to be sure if the error was induced by the new method or by the reference classifier. Of course, this also depends on how challenging the target application is to be classified and on the composition of the traces that are being analyzed. The *GTVS* solution proposed by Canini et al. [2009] is also a strong tool that can significantly improve the task of ground truth identification. Nonetheless, it is based on a combination of different methods to identify the traffic, including DPI, and thus may have similar limitations. On the other hand, although the manual verification of the traffic could allow a better accuracy, it is only

feasible for small datasets. The same happens with the approaches that save information during traffic capturing about the process that generated the packets, as the ones from Gringoli et al. [2009] and Szabó et al. [2008]. This information is very valuable for traffic classification and can help to achieve high accuracy on the ground truth. Unfortunately, the deployment of the client daemons or drivers in all the computers of a large network is also difficult to accomplish. The use of testbeds with smaller networks, in which it is possible to control the applications being used, also allows high accuracy on the ground truth identification. However, it may not be representative of the traffic in large scale networks. The method used to assess the ground truth is extremely important to the quality of the evaluation results. Therefore, one should be aware of the capabilities and limitations of each method when evaluating a classifier.

4.6. Performance Evaluation Metrics

The evaluation of classification methods is made by comparing the results of the classification with the ground truth information of the traces. Each individual case is considered a True Positive (TP), True Negative (TN), False Positive (FP), or False Negative (FN) case depending on whether it was correctly classified as belonging to, correctly classified as not belonging to, incorrectly classified as belonging to, or incorrectly classified as not belonging to a certain class.

The analysis of TPs, TNs, FPs, and FNs can be made in terms of packets, flows, or bytes. The evaluation of classification methods based on packets usually presents lower performance as many packets are similar independently of the application that generate them. For example, a TCP *SYN* packet, used to initiate a connection, is similar for any application. Moreover, many classifiers, especially the ones based on classification *in the dark*, are not design to classify individual packets. The evaluation in terms of flows and bytes may also present different performance levels. In many traces, depending also on the type of traffic they contain, a small number of flows may carry almost all the bytes. The rest of the flows contain only a few small packets. In these cases, if a method correctly classifies only the larger flows, the result of the performance will be very positive in terms of bytes and very negative in terms of flows. On the contrary, if the larger flows are misclassified and the all the small flows are correctly classified, the performance will be positive in terms of flows and negative in terms of bytes.

The performance of the classifiers can be measured, in terms of TPs, TNs, FPs, and FNs, using different metrics [Makhoul et al. 1999; Olson and Delen 2008]. There is a great number of metrics for classification evaluation and, although some are equivalent, most of them measure different aspects of the classification. When using metrics to evaluate a traffic classification mechanism, it is important to understand what is measured by each of them. In the following paragraphs, we briefly explain the most common metrics in traffic classification studies.

The accuracy of a classifier is usually evaluated by measuring its capability to correctly identify positive and negative cases. Hence, *accuracy* is defined as the ratio of correct positive and negative classifications to all the positive and negative cases in the experimental data:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad . \tag{1}$$

However, *accuracy* is insufficient to evaluate a classifier when using imbalanced datasets with a greater number of positive or negative cases in the dataset, as it gives more importance to the most popular class in the dataset. In such case, if a classifier privileges the class with more cases in the dataset, it will always achieve a good *accuracy*. For example, in an extreme case, a completely useless classifier that classifies as *positive* every case in the dataset will achieve a high *accuracy* in a dataset containing,

e.g., 90% of positive cases. Therefore, it is necessary to use more than one measure, each of them evaluating different aspects of the results.

Two of those metrics, *precision* and *recall*, are used together to evaluate classification methods and are defined as follows [Nguyen and Armitage 2008b]:

$$Precision = \frac{TP}{TP + FP} \ , \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \ . \tag{3}$$

Some authors also used the term *accuracy* to refer to *precision* [Callado et al. 2010] or to *recall* [Hu et al. 2008]. These metrics are used to evaluate the capability of the classifier to correctly identify the positive cases. *Precision*, also referred as *positive predictive value*, evaluates how correct the cases identified as positive by the classifier are, whereas *recall*, also referred as *hit rate* or *true positive rate*, expresses the percentage of positive cases included in the dataset that were correctly identified by the classifier.

Nonetheless, *precision* and *recall* also have limitations in specific contexts as they do not value rarity [Weiss 2004; Stefanowski and Wilk 2009]. Both metrics do not consider the amount of negative cases correctly identified by a classifier. This means that if a classifier *C1* returns, e.g., 10 false positives out of 10 negatives and a classifier *C2* returns an equal number of false negatives and of true positives and 10 false positives out of 1000 negatives, both classifiers will have the same *precision* and *recall*. However, *C2* may be considered to have better performance as it failed to correctly identify only 1% of the negative cases, while *C1* was not able to identify any negative case. Furthermore, the precision obtained for a dataset containing an extremely low share of positive cases may be affected by the high prevalence of negative cases. In fact, in such context, a very small percentage of the negative cases misclassified as positive cases may be sufficient to overcome the number of true positives identified by the classifier, due to the shortage of positive cases in the dataset.

In these situations, it may be advantageous to consider metrics that separately evaluate the classification of positive and of negative cases. Therefore, *recall* can be used together with a another metric, *specificity*, which is defined as follows [Wang 2008]:

$$Specificity = \frac{TN}{FP + TN} \ . \tag{4}$$

When used together with *specificity*, *recall* is usually called *sensitivity* [Raahemi et al. 2008b]. *Sensitivity* measures the ratio of correctly classified positive cases to the total of positive cases, whereas *specificity* evaluates the negative cases that were correctly classified. In the context of traffic classification, *sensitivity* and *specificity* are especially useful to evaluate classifiers that are focused on a specific class that accounts for a minority of the traffic in a dataset, e.g., a classifier designed to identify video streaming or Voice over Internet Protocol (VoIP) traffic.

Moreover, Karagiannis et al. [2005a] defined a different metric similar to *recall*, which they called *completeness*, and used it together with *precision*, which they called *accuracy*. To the best of our knowledge, the two metrics were also used by Callado et al. [2009; 2010] and Szabó et al. [2007]. *Completeness* measures the ratio of classified positive cases, correctly or incorrectly, to the total number of positive cases and is defined as follows:

$$Completeness = \frac{TP + FP}{TP + FN} \ . \tag{5}$$

The metrics used to evaluate a classifier should be chosen depending on the context and purpose of each classifier. Although some authors have used different names for

Table II. Well known port numbers used by several P2P protocols.

Protocols	TCP Ports	UDP Ports
AIM - messages	5190	5190
AIM - video	1024–5000	1024–5000
ARES Galaxy	32285	32285
BitTorrent	6881–6999	
Blubster	41170–41350	41170–41350
Direct Connect	411, 412, 1025–32000	1025–32000
eDonkey	2323, 3306, 4242, 4500, 4501, 4661–4674, 4677, 4678, 4711, 4712, 7778	4665, 4672
FastTrack	1214, 1215, 1331, 1337, 1683, 4329	
Gnutella	6346, 6347	6346, 6347
GoBoogy	5335	5335
HotLine	5500–5503	
ICQ	5190	
iMesh	80, 443, 1863, 4329	
IRC	6665–6669	
Kazaa	1214	1214
MP2P	10240–20480, 22321, 41170	41170
MSN	1863	
MSN - file transfer	6891–6900	
MSN - voice	6901	6901
Napster	5555, 6666, 6677, 6688, 6699–6701, 6257	
PeerEnabler	3531	3531
Qnext	5235–5237	5235–5237
ROMnet	6574	
Scour Exchange	8311	
ShareShare	6399	6388, 6733, 6777
Soribada	7675–7677, 22322	7674, 22321
SoulSeek	2234, 5534	2234, 5534
WASTE	1337	1337
WinMX	6699	6257
XMPP / Jabber	5222, 5269	5222, 5269
Yahoo - messages	5050	
Yahoo - video	5100	
Yahoo - Voice	5000–5001	5000–5010

similar metrics, in the next section, we will use the terms *accuracy*, *precision*, *recall*, *sensitivity*, *specificity*, and *completeness* as described above, so as to keep the article coherent.

5. DISCUSSION OF THE STATE OF THE ART ON TRAFFIC CLASSIFICATION

In the literature on traffic classification, several mechanisms and applications are proposed for the identification of application-level protocols. The following subsections provide a theoretical study of the most relevant works in this field of study. Traffic classification methods are, in many cases, suitable for the identification of traffic from different types of applications. Nonetheless, since most of the applications whose traffic is difficult to identify by conventional means use P2P platforms, many of the studies discussed herein are oriented for the detection and classification of P2P traffic. Although the approaches used by most of the studies described in this section are, in many cases, also used for the detection of traffic anomalies, virus, and other software threats [Lakhina et al. 2005; Ranjan et al. 2007; Soewito et al. 2009], this section will be focused only on the studies addressing the subject of traffic classification.

5.1. Port-Based Classification

As described in section 4.1, the early strategies for traffic classification were based on the identification of port numbers. The Internet Assigned Numbers Authority (IANA)

keeps an updated list of the well known or registered port numbers, which is available in the web [IANA 2011]. Nevertheless, there are also port numbers or ranges that are traditionally used by some P2P systems. Table II presents a list of the port numbers commonly used by well known P2P applications.

A few studies have used this approach to identify application protocols. In [Sen and Wang 2004] and in [Krishnamurthy and Wang 2002], the authors analyzed P2P traffic collected at the border routers of a large ISP. In order to distinguish the flows from *Gnutella*, *FastTrack* and *Direct Connect*, they used the TCP port numbers. Saroiu et al. [2002a] collected traffic from the University of Washington and, using port numbers, identified and analyzed the data from four content delivery systems, HTTP web traffic, *Akamai* network, *KaZaA* and *Gnutella*. Leibowitz et al. [2002] monitored traffic from an ISP network and analyzed *FastTrack* based traffic, which includes *KaZaA*, *Morpheus* and *Grokster* data, filtered through the use of port numbers. Gerber et al. [2003] resorted to port numbers, as well, to identify traffic from several P2P systems. They collected traffic from an ISP backbone and from a university network, and analyzed its properties.

There are also tools for traffic analysis that provide information about the application-level protocol based on port numbers, like the *CoralReef* suite [Moore et al. 2001] or the *Wireshark* [2010] packet analyzer. A few studies have used the application port tables from *CoralReef* to identify the network traffic [Fraleigh et al. 2003].

5.2. Deep Packet Inspection Classification

The lack of effectiveness of the methods based on port numbers motivated the increase of the studies that analyze the traffic using payload inspection. Sen et al. [2004] proposed payload signatures for *Gnutella*, *eDonkey*, *Direct Connect*, *BitTorrent*, and *KaZaA*, and implemented them using the *Gigascope* monitor. They tested the solution using traffic collected on an access network to a major backbone and on a T3 (45 Mbps) link connecting a Virtual Private Network (VPN) to the Internet. The authors estimated that the false positives rate was approximately 0%, while the false negatives rate was between 0.00% and 4.97% for the analyzed protocols, with the exception of *BitTorrent* for which it was 9.90%. However, they considered that the flows that use well known port numbers of P2P applications are, in fact, P2P traffic. Based on that assumption, each flow that used one of those ports and was not classified as P2P traffic was identified as a false negative case.

Moore and Papagiannaki [2005] presented a flow-based methodology that resorts to the deep inspection of the payloads. It uses a set of distinct methods that search for known signatures within the full payload of each packet. The methods are checked sequentially until one of them matches a certain application. In the tests performed by the authors, which relied on manual verification, the proposed set of methods was able to accurately identify approximately 99.99% of the traffic, which corresponds to the recall rate.

In [Karagiannis et al. 2004b], the authors used payload signatures to identify traffic of several P2P applications, namely, *eDonkey2000*, *FastTrack*, *BitTorrent*, *WinMX*, *Gnutella*, *MP2P*, *Soulseek*, and *Direct Connect*. They used their approach to analyze a few traffic traces captured from links of two backbones, and conclude on the evolution of the percentage of P2P traffic in the Internet.

Spognardi et al. [2005] collected and analyzed traffic from *OpenNap*, *WPN* and *FastTrack* P2P protocols in order to identify payload signatures. The signatures were codified in *form rules* for the *Snort* NIDS and used to monitor network traffic.

In [Choi and Choi 2006], the use of port numbers is proposed as a real-time method to identify the traffic. Afterwards, the traffic is also inspected off-line using DPI techniques. The authors presented a methodology to check if the packets match a data

pattern that is based on an edit distance algorithm. Bin et al. [2007] proposed a solution that uses payload signatures to identify P2P flows as well. Each successfully identified packet is added to a table with an hash identifier, which is calculated from the source and destination IP addresses and from the transport port numbers. This way, the authors only examine the contents of the packets that belong to flows that were not classified yet.

The detection of chat related traffic was studied in [Dewes et al. 2003]. The authors analyzed several chat protocols and identified payload signatures. The tests show that the methodology presented, which was validated using manual verification, failed to detect less than 8.3% of all chat connections (recall of 91.7%) and, from the ones detected, 93.13% were correctly classified (precision).

Generally, one the major drawbacks of DPI methods is their weight in terms of computation power. Hence, a few studies have tried to develop DPI mechanisms that are light and scalable. Risso et al. [2008] presented a taxonomy of the possible DPI approaches and performed a comparison of the performance and accuracy between a lightweight and a completely stateful traffic classification methods. They concluded that, although the lightweight methods are not so accurate, they are still effective enough for the purpose of traffic classification while being able to perform much faster than the stateful approaches. Guo and Qiu [2008] proposed a signature-based method to identify P2P flows in high-speed networks using packet sampling and they tested it with *BitTorrent*-related traffic. They evaluated the relation between its performance and the sampling probability, achieving different false positive and negative rates, depending on the value of the sampling probability, from 0.00% to 11% and from 0.33% to 10.5%, respectively. In [Cascarano et al. 2009], the authors evaluated the computational cost of a DPI mechanism by comparing it with a statistical one. Although the comparison has been made between only two specific methods, it shows that, depending on the composition of the traces, the DPI mechanism can be as much computationally heavy as the statistical classifier; or it can go as high as five times the complexity of the statistical approach. In her PhD thesis, Yu [2006] developed high speed packet processing algorithms, proposing the use of hardware support to perform the deep inspection of packets. Smith et al. [2008] used auxiliary variables and optimizations to implement a mechanism for deflating explosive Deterministic Finite Automata (DFA). Using their solution, the authors were able to optimize the process of signature matching, achieving promising results for File Transfer Protocol (FTP), SMTP, and HTTP traffic. Kumar et al. [2006] introduced a new representation for regular expressions, called the Delayed input DFA (D^2FA), which significantly reduces the space requirements of a DFA. The results of their tests showed that they were able to reduce memory space requirements by more than 95%. In [Cascarano et al. 2010a], the authors presented two optimizations of a DPI classifier that reduce the data checked by the pattern matching engine. The improvements are achieved at the cost of a controlled reduction of the accuracy, which, unlike the case of intrusion detection, is acceptable in traffic classification.

The encryption of the payload is usually a problem for the DPI techniques. However, a few studies used the payload examination to identify P2P encrypted traffic. In [Carvalho et al. 2009b], the authors identified, manually, several payload signatures of *BitTorrent* encrypted traffic and provided a set of *Snort* rules to match the patterns observed. They tested the rules with traffic from a university network. The same authors have used a similar approach to identify signatures for encrypted *eDonkey* traffic [Freire et al. 2009] and P2P TV traffic [Carvalho et al. 2009a].

Most DPI mechanisms are based on signature matching. Nevertheless, a few methods use the payload data in a different perspective. Dhamankar and King [2007] used entropy to explore the randomness of the data within the encrypted payloads of *Skype*

traffic, resorting to clustering methods and congregating several heuristics. More studies have also addressed the subject of *Skype* traffic identification. Ehlert and Petgang [2006] described a detailed analysis of the *Skype* protocol and presented a signature to detect its traffic that is based on payload and transport-level data.

Some authors have been developing studies on the automatic identification of payload signatures. Most of those studies are focused on the identification of worms, virus, and other traffic anomalies [Singh et al. 2004; Yegneswaran et al. 2005; Cavallaro et al. 2008]. However, a few authors have proposed similar approaches for traffic classification. Haffner et al. [2005] used three ML algorithms and, with two of them, they were able to construct signatures, with precision between 99% and 100% and recall between 86.6% and 99.9%, by resorting to the examination of a small amount of data at the beginning of the communication. The study was performed for traffic from FTP, SMTP, Post Office Protocol (POP), Internet Message Access Protocol (IMAP), HTTP, Hypertext Transfer Protocol Secure (HTTPS), and Secure Shell (SSH). Finamore et al. [2009] presented *KISS*, a classifier that automatically extracts signatures from a UDP stream by using a stochastic test that allows the identification of the application protocol syntax, while ignoring the synchronization and semantic rules. The signatures can be seen as statistical fingerprints in the payload data. The authors tested the mechanism, verifying it manually, using traffic traces from an italian ISP. *KISS* correctly identified more than 98.1% of the samples in the worst case, reaching an average recall of 99.6% and an average false positives rate of 0.34%. In [Mantia et al. 2010], they extended the previous method to support also the classification of TCP traffic, with an average recall of 97.62%. In [Park et al. 2008], it is also presented a solution for the automated creation of signatures, the *LASER* algorithm. The authors tested the approach for *LimeWire*, *BitTorrent*, and *Fileguri*, using data collected in a campus network and manually verified. They achieved an accuracy rate of 97.39%, with a false negatives rate of between 0.39% and 10.40% and with 0% of false positives.

5.3. Classification *In The Dark*

Recently, several studies have proposed classification strategies that rely on behavioral and statistical patterns, which can be further categorized as follows.

5.3.1. Heuristics. Several studies propose heuristics as a means to identify P2P traffic. Constantinou and Mavrommatis [2006] proposed a classifier that uses three heuristics: the number of hosts that act both as server and client in a specific port exceeds a given threshold; the estimated network diameter is at least as great as 2; and the number of hosts that are present in the first and last levels of the network exceeds a given threshold. The method was tested using data traces from *NLANR* [2010] and compared with port-based classification. Depending on the threshold values, the results vary between 8.5% and 12.7% of false negatives (detected with port-based and not detected with heuristics) and between 7.6% and 42.4% of additional positives (not detected with port-based and detected with heuristics). In [Perényi et al. 2006], the authors described a method based on a set of six heuristics to identify P2P traffic: simultaneous usage of TCP and UDP; the existence of several consecutive connections between two hosts; well known P2P port numbers; multiple flows with the same flow identities; an IP using the same transport port more than 5 times in the measurement period; and the flow size larger than 1MB or its duration longer than 10 minutes. The validation of the approach was made using a small labeled traffic trace and it achieved a recall of 99.14% and of 97.19% for P2P and non-P2P traffic, respectively, with 0.3% of false positives and 0.8% of false negatives. John and Tafvelin [2008] also proposed a set of heuristics to classify Internet traffic, which are a redefined combination of the ones suggested in [Karagiannis et al. 2004c; Perényi et al. 2006]: the concurrent use of TCP

and UDP; the well known P2P port numbers; the port numbers that are used very often; the relation between the number of IP addresses and the number of transport ports; and the flows carrying more than 1 MB or lasting more than 10 minutes. Besides the heuristics, the authors also described a set of rules to reduce the number of false positive cases. They used the mechanism to classify traces collected at a university link, leaving only 2% of the traffic unclassified (recall of 98%).

5.3.2. Social Behavior. Karagiannis et al. [2005a] presented *BLINC*, a mechanism for flow classification that does not rely on the payload data or transport port numbers to identify the application protocol. *BLINC* analyzes traffic at three levels (social, functional, and application) exploiting properties of each node, like the relation with the remaining hosts, the role in the connection (server or client), the transport layer information, or the average packet size. The mechanism was tested using traffic collected at numerous academic, research and residential complexes, within a university campus and it was evaluated by comparing it with a DPI based method. *BLINC* was able to classify between 80% and 90% of the flows, corresponding to the completeness rate, with a precision ranging from 90% to 95%. Iliofotou et al. [2007] introduced a different perspective for the traffic analysis that is focused on the network-wide interactions of hosts. They model the social behavior of hosts by organizing and correlating the information in graphs, which they call Traffic Dispersion Graphs (TDGs), where the edges represent different interactions. In [Iliofotou et al. 2008; Iliofotou et al. 2009], they used TDGs to create a framework, *Graption* (Graph-based classification), to classify the traffic based on the application protocol. The mechanism was tested using two traces from a Tier-1 ISP and one trace from the *Abilene* (*Internet2*) network and a DPI-based method as baseline. The results showed that the solution was able to classify the traffic with a recall of between 94% and 95% and a precision of between 95% and 96%.

5.3.3. Statistical or Behavioral Signatures. A mechanism for flow classification based on the definition of statistical signatures or fingerprints for different traffic classes was proposed by Crotti et al. [2006; 2007]. The fingerprints are created using traffic pre-classified with any effective mechanism and then used to classify network traffic. Dusi et al. [2008; 2009] also used statistical fingerprints to identify encrypted tunnels. The method was evaluated, using data collected on controlled sessions and reaching a recall of between 82.45% and 100.00%. Bartlett et al. [2007a] identified three basic behavioral signatures from P2P file sharing: failed connections, the ratio of incoming and outgoing connections, and the use of unprivileged ports. They evaluated the mechanism by classifying *BitTorrent* and *Gnutella* traffic, captured from a commercial ISP and from academic institutions. In order to access the ground truth for *BitTorrent* data, the authors identified all the flows that used the default port number of *BitTorrent* tracker and manually verified that the destination was a real tracker. All the traffic identified by these means was confirmed to be P2P traffic. Additionally, they considered all flows using non-privilege ports that are not well-known ports as *likely non-P2P*. For the *Gnutella* data, the authors considered as P2P all the flows that contact with *Gnutella* ultra-peers, which they identified by connecting repeatedly to the *Gnutella* network and keeping a record of the ultra-peers list. These approaches were used to verify the classification of P2P hosts and of *likely non-P2P* hosts. Besides of this strategies, to verify the classification of the remaining flows, the authors identified the flows using default *BitTorrent* ports (6969, 6881-6888) and the default Gnutella port (6346). The results show that *BitTorrent* hosts were detected with a recall ranging from 83% to 92%, while *Gnutella* hosts achieved a recall from 57% to 97%, and the false positives rate was between 2% and 25%. In [Freire et al. 2008a; 2008b], a mechanism to identify VoIP calls hidden in Web traffic was proposed. The authors analyzed several properties of the network data to distinguish between VoIP and *legitimate* Web traffic and

selected the following parameters: Web request size; Web response size; inter-arrival time between requests; number of requests per page; and page retrieval time. In order to measure the *goodness of fit*, they used the *Chi-square* and *Kolmogorov-Smirnov* tests. The evaluation was made using *Skype* and *Google Talk* VoIP data, previously collected in both ISP and university links on a controlled way. The method achieved similar results for both protocols, being able to identify around 90% (recall rate) of the VoIP calls with a false positives rate of 2%, and 100% (recall rate) of VoIP calls hidden in Web traffic with a false positives rate of 5%. In [Gomes et al. 2008], the authors analyzed several edge user traces of P2P and non-P2P applications and tried to identify a behavioral pattern of the P2P traffic. They concluded that the packet sizes of P2P traffic presented a high heterogeneity when compared to the packet sizes of the non-P2P traffic. They used entropy to represent the heterogeneity degree and calculated its value for a sliding window containing a fixed number of packets. P2P traffic, especially the one related with VoIP services, returned high entropy values, while for the regular *client-server* traffic the entropy value was consistently smaller. Lin et al. [2009] proposed the use of the packet size distribution and port association as a pattern to distinguish application protocols. They used traces collected in a controlled environment to evaluate the method, which achieved a recall of between 74% and 100% and false positives and negatives rates ranging from 0% to 9% and from 0% to 18%, respectively. Palmieri and Fiore [2009] presented a new approach for the classification of Internet traffic that relies on Recurrence Quantification Analysis (RQA). They studied non-linear properties of specific IP flow types so that they could determine the recurrence phenomena and hidden non-stationary transition patterns related to each type of traffic. For the different traffic classes considered in the study (HTTP, *eDonkey2000*, Domain Name System (DNS), SMTP, POP, and SSH), the authors obtained average recall rates ranging from 45.8% to 89.4%, when compared to DPI. The tests were performed with three distinct traces captured in a university network.

5.3.4. Machine Learning Algorithms. The supervised and unsupervised ML methods are widely used in studies on the classification of network traffic. In the following paragraphs, the different research works based on ML are organized depending on the techniques employed.

Naïve Bayes and Neural Networks. A Naïve Bayes estimator was employed in [Moore and Zuev 2005; Zuev and Moore 2005] to distinguish the traffic based on the application-level protocol and they used hand-classified data to train the classifier. The input discriminators for this study were formed by several properties of the flows. The method was tested with traffic data from a research campus, previously hand-classified and collected twelve months later than the data used for the training process (which proves the temporal stability of this approach), and achieved a precision of between 13.46% and 99.27% and a recall of between 93.73% and 96.29%. Schmidt and Soysal [2006] proposed a mechanism for the detection of P2P traffic resorting to a Bayesian network, built using the following flow characteristics: IP packet size distribution; packets per flow distribution; octets per flow distribution; flow time distribution; and well-known port numbers. They evaluated the performance of the classifier using traffic from an academic network and compared the results against a signature-based method. The results showed false negatives and positives rates ranging from 16% to 26% and from 22% to 28%, respectively. Auld et al. [2007] also described a classifier based on a Bayesian neural network, trained using data previously classified using DPI. A set of traffic properties and statistics was used as input for the classification process. The method proved to have an accuracy of between 95% and 99%, for data manually verified and collected eight months after the data used to train the classification mechanism.

Clustering. McGregor et al. [2004] proposed a clustering-based methodology that extracts a range of flow properties and uses the Expectation-Maximization (EM) algorithm to cluster the flow into different classes. A preliminary validation of the approach showed promising results. A framework for traffic classification, based also on the EM algorithm and trained using several flow characteristics, was described in [Zander et al. 2005a; 2005b]. The method was tested using traffic traces from *NLANR* [2010] and the results showed moderate effectiveness. Nguyen and Armitage [2006; 2008a] proposed a solution based on the EM algorithm and on a Naïve Bayes classifier. In order to test and classify the method, they used traffic from a gaming server and from a university link and obtained its ground truth using the port numbers. The results showed an average accuracy above 98.3%. Bernaille et al. [2006a; 2006b] presented a method for traffic classification that is based on the first five packets of a TCP connection, excluding the control packets (the ones marked with the flags *SYN*, *ACK*, etc.). They experimented three clustering techniques to explore the relations between the initial packets and identify clusters related with distinct application protocols: *k-means*, Gaussian Mixture Model (GMM), and spectral clustering. The mechanism was trained using an one-hour trace collected at the edge of a university network and it was tested with a similar trace captured six months later, by comparing it with a DPI-based classifier. The results presented a recall of between 36.0% and 100.0% and a false positives rate from 0.0% to 3.6%. In [Bernaille and Teixeira 2007], the authors extended the same approach, using GMM to identify traffic encrypted (or tunneled) in Secure Sockets Layer (SSL) connections. The evaluation, performed using manually generated traffic traces, showed a recall ranging from 81.20% to 100.00% and a false positives rate of between 0.00% and 2.30%. Erman et al. [2006a] also described a preliminary work on the effectiveness of clustering algorithms for traffic classification. They employed the *k-means* and the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithms and used several properties to discriminate the flows, like the total number of packets, the mean packet size, the mean payload size, the number of bytes transfered, and the mean inter-arrival time. The approach was tested using a publicly available trace without the payload data and a trace collected by the authors at a university link, showing a recall ranging from 86.6% to 93.5%. The ground truth verification was made using port numbers and DPI. In [Erman et al. 2006b], they proposed an unsupervised ML solution, the EM algorithm, for the traffic classification. They analyzed the performance of the method using traffic traces collected at a university link and compared the results with a supervised ML technique, a Naïve Bayes classifier. The evaluation showed that the EM algorithm achieved precision and recall rates between 80% and 100%. The same authors [Erman et al. 2007a] proposed also a semi-supervised learning method for traffic discrimination, based on several flow-related statistics, that allows the classifiers to be designed from training data formed by a few labeled and many unlabeled flows. Although the mechanism is not limited to any specific clustering algorithm, after the previous studies, they decided to use the *k-means*. They tested the mechanism using data whose ground truth was verified using DPI, heuristics and manual verification, and achieved a recall of between 80% and 90%. The same approach was also used in [Erman et al. 2007b] to distinguish between Web and P2P traffic, with an accuracy of between 80% and 95%, precision of between 71.42% and 97.84%, and recall of between 62.10% and 97.32%.

Decision Trees. A method for traffic classification based on decision trees was proposed in [Early et al. 2003]. The trees were constructed by employing the *C5.0* algorithm and using the information about the TCP flags used in each connection, as the authors believe to be enough to capture the flow behavior. The authors used HTTP, FTP, *Telnet*, SMTP, and SSH traffic to test and evaluate the mechanism, which proved

to have a recall of between 82% and 100%. Cao et al. [2008] described an approach for the identification of application protocols in real-time, at both host and flow levels, using Classification And Regression Tree (CART). Through an off-line analysis, they extracted metrics to characterize the traffic and used decision trees to identify the traffic in an on-line manner. The authors focused their experiments on traffic from *BitTorrent*, HTTP, SMTP, and FTP, collected in a home link and also in an enterprise network. In order to assess the ground truth, the authors created the traces of *BitTorrent* actively, in a controlled manner, at a home environment. The HTTP, SMTP, and FTP traffic was captured in an enterprise network and filtered using the port numbers. In the tests the authors performed, the method classified the traffic with a false positives rate of between 0.05% and 12.7%, and a false negatives rate of between 0% and 17.9%. Raahemi et al. [2008b] applied Concept-adapting Very Fast Decision Tree (CVFDT) to identify P2P traffic, using a set of network level attributes of the packets. They used labeled data sets to evaluate the performance of the mechanism, achieving an accuracy of between 79.50% and 98.65%, a specificity of between 82.96% and 95.89%, and a specificity of between 67.96% and 99.72%. In [Angevine and Zincir-Heywood 2008] the *C4.5* and the *AdaBoost* algorithms were used to classify UDP and TCP Skype flows. The authors used the mechanism to analyze labeled traffic traces from a university network with a recall between 94% and 99% and a false positives rate between 1% and 26%. A decision tree based classifier, *Random Forests*, was used in [Wang et al. 2008] to identify traffic from multiple P2P protocols. The method was tested with manually labeled datasets, captured at residential and academic networks and achieved an accuracy rate ranging from 89.38% to 99.98%, a precision from 32.69% to 100.00%, and a false positives rate from 0.00% to 12.61%. Branch et al. [2009] also employed the *C4.5* algorithm using different conjunctions of flow features from packet lengths, statistics of large packets and inter-arrival times. Using traffic from a university network, the method was able to classify the traffic with a precision of 99% and a recall of 98%.

Markov Chains and Models. Wright et al. [2006] focused specifically on the behavior of encrypted traffic. Using a classifier based on hidden Markov models and also on the *k-nearest neighbor* algorithm, they proved that it is possible to identify the application-level protocol: in aggregate traffic without demultiplexing or reassembling the TCP connections; in aggregate traffic by demultiplexing the flows and analyzing them individually; and in aggregate traffic without demultiplexing the flows or recognizing which packets in the aggregate traffic belong to which flows (as when the traffic is encrypted at the network layer). The evaluation was performed using traffic from SMTP, HTTP, HTTPS, FTP, SSH, *Telnet*, and AOL Instant Messenger (AIM) and the ground truth information was verified using port numbers, presenting a recall ranging from 57.70% to 96.70% and a false positives rate of between 0.62% and 8.37%. Dainotti et al. [2008] have also proposed a classification mechanism based on hidden Markov models, whose classification process is based on packet sizes and inter packet times. The authors applied the model to real traffic traces, verified manually and using DPI, of two multi-player games, HTTP, SMTP, *eDonkey*, *PPlive* P2P TV, and *MSN Messenger*, reaching a recall of between 90.23% and 100.00%. Xusheng and Zhiming [2009] used Markov chains to model the sequences of control packets a certain application exchanges with a remote host and based the decision rule on the *Neyman-Pearson* test and on the likelihood criterion.

Support Vector Machines. The behavioral-based classification was accomplished in [Gonzá1ez-Castaño et al. 2006] by employing SVMs. The solution proposed was evaluated using datasets that were labeled based on the port numbers and on a few simple heuristics, reaching an accuracy of between 78.7% and 90.2%. In [Turkett et al. 2008], the authors extracted several flow features and used FTP, SSH, *Telnet*, SMTP,

HTTP, and POP related traffic to train a SVM mechanism, which performed well in the tests they conducted. Este et al. [2008] proposed three pattern recognition solutions based on SVMs, GMM, and *C4.5* to identify the presence of the unknown classes and they used the size of the first packets as input feature. The tests performed with the three methods presented an accuracy of between 92.53% and 98.83%, confirmed using DPI and manual verification. In [Este et al. 2009], the authors described carefully the approach based on SVMs and used it to classify three sets of traffic. The results of the test showed a recall ranging from 69.6% to 100.0%. Valenti et al. [2009] described a new approach to identify the traffic from P2P-TV applications resorting to the number of packets exchanged between the peers during short time intervals and uses SVMs to train the mechanism. The authors captured traffic in a large testbed and used it to test the method, which was able to correctly classify between 91.3% and 99.6% the data (recall rate), with only between 0.3% and 8.7% of false positives. An approach relying on SVMs was also proposed in [Sena and Belzarena 2009]. The authors used the size of the first N packets of each flow as a feature for traffic classification and they trained the mechanism using data previously classified through DPI. They tested the method using traffic from the network of an uruguayan ISP and achieved an accuracy ranging from 30% to 100%.

Other Studies Relying on Machine Learning Techniques. In [Liu et al. 2007], the authors used the ratio between the amount of download and upload traffic, in each minute, as an identification pattern for each application and proposed a supervised ML algorithm to identify each distinct class. They tested the method with traffic from a few P2P applications, namely *Maze*, *BitTorrent*, *PPlive*, *eDonkey*, and *thunder*, which they collected on a testbed. The results showed an accuracy rate of between 78.5% and 99.8%, depending on the protocol. Raahemi et al. [2008a] employed Fuzzy Predictive Adaptive Resonance Theory (ART), or Fuzzy *ARTMAP*, to identify P2P traffic. They used only data from the IP headers to build the Fuzzy *ARTMAP* neural networks. The experimental tests, using labeled datasets, showed that the classifier is able to perform with an accuracy ranging from 78% to 92%, a sensitivity from 68% to 90%, and a specificity from 85% to 96%. Huang et al. [2008] used a set of discriminators from which they identified patterns by resorting to a ML technique. In this work, the authors experimented a few techniques, concluding that Bayesian network, Partial Decision Tree (PART), and *C4.5* are the ones that performed better. The evaluation was made using traces collected at a university link, whose ground truth was accessed using payload signatures. The method showed a recall of between 90.87% and 95.11%, depending of the ML technique used. Hu et al. [2008; 2009] proposed a novel method for the classification of P2P traffic that aims to build behavioral profiles for each application, by using *association rule mining*. They choose five flow tuples, extract flow statistics, and correlate them using the *Apriori* algorithm. The approach, which was tested for *BitTorrent* and *PPLive* using on-campus traces, verified manually and through DPI, presented a recall ranging from 90.0% to 98.0% and a false positives rate between 0.2% and 5.0%. In [Williams et al. 2006], the authors compared five ML algorithms for traffic flow classification. They argued that it is useful to analyze the algorithms in terms of computational efficiency rather than classification accuracy as, even though the accuracy between distinct algorithms may be similar, the computational efficiency can be considerably different. Based on their results, the authors concluded that *C4.5* algorithm was able to classify the flows faster then the remaining algorithms. A similar conclusion was reached in [Soysal and Schmidt 2007], in which three solutions for P2P flows detection, based on ML, were compared.

5.3.5. Service Identification. Baldi et al. [2009] described a new approach for traffic classification that relies on the identification of the service that generates the traffic. They

defined *service* as a triple formed by IP address of the server, transport port at the server, and transport protocol. The authors say that the method can be seen as a complement to reduce the computation and memory requirements of the existing solutions. Nevertheless, in the tests performed on the Internet link of a university campus, the mechanism was able to successfully classify 81% of the packets and 93% of the data (recall rate).

5.4. Classification Based on Active Mechanisms

Although active methods are especially suitable for network performance studies, they can also be used on traffic detection mechanisms. The *Napster* and the *Gnutella* systems were analyzed in [Saroiu et al. 2002b; 2003] with the purpose of characterizing the population of end-user hosts. The authors of the study created a crawler for each of the systems that gathered information regarding different properties, like bottleneck bandwidths, IP-level latencies, etc. As the goal of the study was to characterize both systems, the results presented are not focused on the classification accuracy but in the properties of the traffic.

Ohzahata et al. [2005] have also proposed an active approach to identify pure P2P traffic and applied their methodology to the *Winny* P2P file-sharing system. They developed a crawler to collect information of the IP addresses and transport ports of the hosts connected to the system and they used it to identify further peers. The study provides results regarding the number of peers identified by the mechanism, but its accuracy was not evaluated.

5.5. Classification Through the Combination of Approaches

A few studies propose solutions that combine different kinds of approaches for the classification of network traffic. Karagiannis et al. [2004c] proposed a cross-validation mechanism which uses port numbers, payload signatures, and behavioral patterns to identify traffic from *eDonkey*, *Fasttrack*, *BitTorrent*, *Gnutella*, *MP2P*, *Direct Connect* and *Ares*. Besides presenting payload signatures for the said applications, the authors propose a non payload based method that uses two heuristics. The first heuristic identifies source-destination IP pairs that use both TCP and UDP. The second one says that, when the number of distinct IP addresses connected to a destination IP is equal to the number of distinct ports used for the connections, the flows are likely to be related to P2P applications. Their behavioral mechanism identified more than 90% of the total P2P bytes and 99% of the P2P flows, which corresponds to the recall rate. The false positives rate, which was calculated by comparing the results of the payload mechanism with the results of the behavioral method, is of approximately 8% to 12% of the total estimate of P2P traffic. Nevertheless, the authors argue that part of the false positives are, in fact, true positives that were not identified by the payload based mechanism.

Dedinski et al. [2005] presented an architecture for the detection and control of P2P traffic. It makes use of active crawlers to collect information about the peers of a specific application and, this way, understand the topology of the correspondent overlay network. Alongside, the network-level properties of the traffic are also analyzed, either per-packet or inter-packet, and used as a behavioral pattern, which the authors identify using wavelet analysis techniques. They performed a preliminary test of the architecture using only *eDonkey* and FTP traffic.

In [Nogueira et al. 2007; Couto et al. 2008; Nogueira et al. 2009], a framework to identify Internet applications using a neural networks-based method was proposed, relying on a previous identification obtained through any alternative technique. The authors also described a module to classify the traffic using payload signatures that was employed in the training of the neural network. They tested the method for traffic

from *BitTorrent*, *eMule*, *Gnutella* and HTTP, collected individually for each application, achieving a recall between 90% and 99%.

Bonfiglio et al. [2007] proposed a Naïve Bayes classifier based on two traffic characteristics, the message size and the average inter packets gap. They also implemented a classifier based on the packet payload that uses the *Chi-Square* test to identify *Skype* traffic by exploiting the randomness induced by the encryption of the payload. The authors tested their approach using traffic from an ISP and from a campus and compared its accuracy against a signatures based method, reaching a false positives rate between 0.00% and 2.40% and a false negatives rate between 2.96% and 29.98%.

A mathematical framework for unsupervised protocol inference was described in [Ma et al. 2006]. The authors introduced three methods for identifying different aspects of the communication of a certain protocol: product distributions of byte offsets; Markov models of byte transitions; and common substring graphs of message strings. They evaluated the mechanism using traces collected at different buildings of the university campus and verified manually. Depending on each traffic class in the different traces, the precision was between 68.81% and 100.0% for product distributions, between 0.0% and 100.0% for Markov models, and between 76.87% and 99.99% for common substring graphs. The recall was between 81.82% and 100.0%, 0.0% and 100.0%, and 48.76% and 100.0%, for product distributions, Markov models, and common substring graphs, respectively.

Szabó et al. [2007] presented an architecture that can be extended with modules for distinct traffic classification approaches. They analyzed the performance of the solution using traffic captured in the network of five mobile operators in Europe and Asia. The effectiveness of the classification was also evaluated using hand-classified data and traces captured in a controlled environment.

Adami et al. [2009] proposed a new algorithm to identify, in real-time, the hosts in a network that are using *Skype* clients, that relies in payload signatures and statistical analysis. The algorithm is able to recognize the different types of *Skype* communication: direct calls, calls using a relay node, call to phone service, and file transfer. It was tested on-line and off-line, using traffic from a university LAN and from a small LAN connected to an ADSL link and its performance was compared with the performance of five other classifiers. The traces used in [Bonfiglio et al. 2007] were also tested. The results showed a percentage of false positives between 0% and 0.01%, and of false negatives between 0.06% and 0.64%, in terms of bytes and flows and for both TCP and UDP traffic. The exception was the false negatives rate for TCP flows, whose value was 27.46%.

Callado et al. [2010] collected five distinct datasets, captured in different contexts. The first one was formed by traces from individual applications captured in client computers, which where assembled in a single dataset. By creating the traces manually, the authors could be sure of the applications that generated the traffic. The other datasets were captured in a laboratory network, in an academic backbone, and in the core router of a commercial link (only one direction). The fifth dataset was formed my the traffic in only one direction of the trace from the academic backbone. The ground truth of this four datasets was obtained using DPI. The authors then used six ML algorithms implemented in *Weka* [Hall et al. 2009] to classify the traffic in the five datasets: *J48* (*C4.5* decision trees), PART, *NBTree* (decision trees with Naïve Bayes classifiers on the leaves), Bayesian networks with simple estimator, Bayesian networks with kernel estimator and SVMs. They concluded that none of the classifiers performed better in all the datasets (which correspond to different contexts) and thus they presented a method to combine different classifiers. In order to choose the result of the combination, they proposed five algorithms: *random selection*, *maximum likelihood combination*, *Dempster-Shafer combination*, and *enhanced Dempster-Shafer combination*.

Although it was tested mostly with ML algorithms, the method is independent of the classifiers one may want to combine. In fact, they have also used *BLINC* [Karagiannis et al. 2005a] and DPI in some of the combinations. The results of the evaluation showed that the precision varies from 60% to 99% and the completeness varies from 90% to 100%, depending on the dataset used. Nevertheless, the lower accuracy values were obtained for the datasets that contained only one direction of the flows.

5.6. Applications for Traffic Classification

Besides the research studies proposing solutions for traffic classification and the commercial tools, there are also a few available and *ready to use* applications, which are described in the following paragraphs.

L7-filter [2010] is a classification tool for Linux *Netfilter* subsystem that uses the application layer data to identify the packets. It is widely used in many studies, being, most of the times, the comparison baseline for the performance evaluation of new methods. Another DPI-based tool is *l7-netpdlclassifier* [2010], which is based on the *NetBee* [2010] library and uses a signature database [NetPDL 2010] written using the *NetPDL* language [Risso and Baldi 2006]. *ipoque* has also made available an open-source version of their DPI tool, which they called *OpenDPI* [2010].

Antoniades et al. [2006] developed *Appmon* [2010], a tool for the application-level classification of network traffic. It is based on two *MAPI* [2010] function libraries and it relies on port numbers and data signatures to identify the protocols.

Dainotti et al. [2009] presented *TIE* [2010], a novel community-oriented software for traffic classification. It uses the *libpcap* library and it supports distinct definitions of sessions and classes, as well as it allows the implementation of additional classification plugins. In its initial state, *TIE* is available with only two classification plugins: port numbers (based on *CoralReef*) and payload signatures (based on *L7-filter*).

A classification tool based on clustering mechanisms, called *NetADHICT* [2010], is proposed in [Inoue et al. 2007]. It decomposes the traffic without the use of any application-specific knowledge and it uses an *AJAX*-based web interface that allows one to see the high-level structure of network traffic.

Although *CoralReef*, *Snort*, and *Wireshark* are not especially intended to classify the traffic regarding the application protocol, they do provide simple mechanisms for such purpose. The *CoralReef* suite gives the user the ability to identify the application-level protocol based on the port numbers; *Snort*, by default, contains several rules to identify signatures in the contents of the packets of several protocols; while *Wireshark* is also able to recognize payload patterns in non-encrypted traffic.

5.7. Summary and Challenges

Tables III, IV and V summarize the textual analysis included in the previous subsections. Furthermore, the chronological ordering of the tables allows one to observe the evolution of the approaches used for the traffic classification and of the protocols each study addressed. In the tables, the *Protocols* columns are related with the protocols considered by each study. The performance metrics included in these tables were chosen as they were the most used ones in the research works described along this survey.

The studies proposing DPI-based solutions are listed in Table III, along with the indication of which were used by their authors to identify encrypted or obfuscated traffic. For the sake of simplicity, the studies that evaluated or compared the performance of existing methods were not included in the table.

Since VoIP traffic raises special concerns for network administrators, a few recent studies have been presenting mechanisms for its detection. Table IV provides a summarized analysis of the approaches they used and their performance.

Table III. Studies based on DPI, and their capability to be applied to encrypted traffic.

Studies	Protocols	Encryption
Dewes et al. [2003]	Chat protocols	Does not apply
Sen et al. [2004]	*Gnutella*, *eDonkey*, *Direct Connect*, *BitTorrent*, *KaZaA*	Does not apply
Karagiannis et al. [2004b]	*eDonkey2000*, *FastTrack*, *BitTorrent*, *WinMX*, *Gnutella*, *MP2P*, *Soulseek*, *Direct Connect*	Does not apply
Moore and Papagiannaki [2005]	multiple protocols	Does not apply
Spognardi et al. [2005]	*OpenNap*, *WPN*, *FastTrack*	Does not apply
Haffner et al. [2005]	FTP, SMTP, POP, IMAP, HTTP, HTTPS, SSH	Apply
Choi and Choi [2006]	multiple protocols	Does not apply
Ehlert and Petgang [2006]	*Skype*	Apply
Bin et al. [2007]	P2P	Does not apply
Dhamankar and King [2007]	multiple protocols	Apply
Guo and Qiu [2008]	*BitTorrent*	Does not apply
Smith et al. [2008]	FTP, SMTP, HTTP	Does not apply
Park et al. [2008]	*LimeWire*, *BitTorrent*, *Fileguri*	Apply
Carvalho et al. [2009a]	P2P TV	Apply
Carvalho et al. [2009b]	*BitTorrent*	Apply
Freire et al. [2009]	*eDonkey*	Apply
Finamore et al. [2009]	multiple protocols	Apply
Mantia et al. [2010]	multiple protocols	Apply
Cascarano et al. [2010a]	multiple protocols	Does not apply

Table IV. Studies addressing the subject of VoIP traffic identification and an overview of their performance, in terms of precision (P), recall (R), false positives (FP), or false negatives (FN).

Studies	Approach	Protocols	Performance (%)
Bonfiglio et al. [2007]	Naïve Bayes and *Chi-Square* test	*Skype*	FP: 0.00–2.40; FN: 2.96–29.98
Angevine and Zincir-Heywood [2008]	*C4.5* and *AdaBoost*	*Skype*	R: 94–99; FP: 1–26
Freire et al. [2008a; 2008b]	behavioral signatures	*Skype* and *Google Talk*	R: 90–100; FP: 2–5
Branch et al. [2009]	*C4.5*	*Skype*	P: 99; R: 98
Adami et al. [2009]	DPI and statistical analysis	*Skype*	FP: 0–0.01; FN: 0.06–27.46

Table V describes the characteristics and performance of most of the studies analyzed in this survey that present methods for classification *in the dark*. For the sake of simplicity, only the studies that proposed a new method and evaluated its performance were included. The *Baseline* column indicates how the ground truth information of the traffic used in the evaluation was assessed, or what method was used as a reference to calculate the accuracy value. Most of the terms used in this column are easily understandable. The expression *manual* refers to traces manually verified or classified, *controlled traces* refers to manually or actively generated traces, and *testbed* refers to traces captured in previously prepared testbeds. This table is not meant to be a comparison between the methods, as the evaluations were made by the authors under different conditions and using distinct metrics [Salgarelli et al. 2007]. Its only purpose is to provide an overview of the behavioral methods presented in the literature.

Likewise, Table VI provides a side-by-side comparison of the different approaches followed by several studies in the literature. In order to keep the table short, we added only a maximum of four studies for each type of method and gave priority to the most recent ones. The evaluation results were included to give an easy perception of the performance of each method. Additionally, another column was added to describe the ability of the method to be applied to traffic with encrypted transport-level payloads. Although the studies based on port numbers did not addressed the encryption issue, we considered them suitable for encrypted traffic since the TCP and UDP port numbers are usually not encrypted.

Table V. Summary of the studies presenting new methods for traffic classification *in the dark* and an overview of their performance, in terms of accuracy (A), precision (P), recall (R), sensitivity (Sens), specificity (Spec), completeness (C), false positives (FP), or false negatives (FN).

Studies	Approach	Protocols	Performance (%)	Baseline
Early et al. [2003]	*C5.0*	HTTP, FTP, *Telnet*, SMTP, SSH	R: 82–100	ports
Karagiannis et al. [2004c]	DPI and heuristics	*eDonkey*, *Fasttrack*, *BitTorrent*, *Ares*, *Gnutella*, *MP2P*, *Direct Connect*	R: 90–99; FP: 8–12	DPI
Karagiannis et al. [2005a]	social behavior	multiple protocols	P: 95–99; C: 80–90	DPI
Moore and Zuev [2005]	Naïve Bayes	multiple protocols	P: 13.46–99.27; R: 93.73–96.29	manual
Constantinou and Mavrommatis [2006]	heuristics	P2P	FP: 7.6–42.4; FN: 8.5–12.7	ports
Wright et al. [2006]	hidden Markov models and *k-nearest neighbor*	SMTP, HTTP, HTTPS, FTP, SSH, *Telnet*, AIM	R: 57.70–96.70; FP: 0.62–8.37	ports
Schmidt and Soysal [2006]	Naïve Bayes	P2P	FP: 22–28; FN: 16–26	DPI
Gonzá1ez-Castaño et al. [2006]	SVMs	multiple protocols	A: 78.7–90.2	ports and heuristics
Perényi et al. [2006]	heuristics	P2P	R: 97.19–99.14; FP: 0.3; FN: 0.8	small labeled trace
Ma et al. [2006]	product dists, Markov models, and substring graphs	multiple protocols	P: 0.0–100.0; R: 0.0–100.0	manual
Bernaille et al. [2006a; 2006b]	*k-means*, GMM, and spectral clustering	multiple protocols	R: 36.0–100.0; FP: 0.0–3.6	DPI
Erman et. al [2006a; 2006b; 2007a; 2007b]	*k-means* and DBSCAN	multiple protocols	A: 80–95; P: 71.42–100.00; R: 62.10–100.00	ports and DPI
Nguyen and Armitage [2006; 2008a]	EM and Naïve Bayes	multiple protocols	P: 98.3–99.7; R: 96.0–98.9	ports
Bernaille and Teixeira [2007]	GMM	multiple protocols	R: 81.20–100.00; FP: 0.00-2.30	controlled traces
Auld et al. [2007]	Bayesian neural networks	multiple protocols	A: 95–99	manual
Liu et al. [2007]	supervised learning algorithm	*Maze*, *BitTorrent*, *PPlive*, *eDonkey*, *thunder*	A: 78.5–99.8	testbed
Bartlett et al. [2007a]	behavioral signatures	P2P file sharing	R: 57–97; FP: 2–25	ports and manual
Nogueira et al. [2007; 2009]	DPI and neural networks	*BitTorrent*, *eMule*, *Gnutella*, HTTP	R: 90–99	individual traces
John and Tafvelin [2008]	heuristics	multiple protocols	R: 98	comparison other methods
Wang et al. [2008]	*Random Forests*	multiple protocols	A: 89.38–99.98; P: 32.69–100.00; FP: 0.00–12.61	labeled traces
Cao et al. [2008]	CART	*BitTorrent*, HTTP, SMTP, FTP	FP: 0.05–12.7; FN: 0–17.9	ports and controlled traces
Dainotti et al. [2008]	hidden Markov models	gaming, HTTP, SMTP, *eDonkey*, *PPlive*, *MSN*	R: 90.23–100.00	DPI and manual
Raahemi et al. [2008b]	CVFDT	P2P	A: 79.50–98.65; Sens: 82.96–95.89; Spec: 67.96–99.72	labeled traces
Raahemi et al. [2008a]	Fuzzy *ARTMAP* neural networks	P2P	A: 78–92; Sens: 68–90; Spec: 85–96	labeled traces
Huang et al. [2008]	Bayesian network, PART, and *C4.5*	multiple protocols	R: 90.87–95.11	DPI
Este et al. [2008; 2009]	SVMs, GMM, and *C4.5*	multiple protocols	A: 92.53–98.83; R: 69.6–100.0	DPI and manual
Iliofotou et al. [2008; 2009]	social behavior	P2P	P: 95–96; R: 94–95	DPI
Dusi et al. [2008; 2009]	behavioral signatures	encrypted tunnels	R: 82.45–100.00	testbed
Hu et al. [2008; 2009]	behavioral signatures	*BitTorrent* and *PPLive*	R: 90.0–98.0; FP: 0.2–5.0	DPI and manual
Lin et al. [2009]	behavioral signatures	multiple protocols	R: 74–100; FP: 0–9; FN: 0–18	testbed
Valenti et al. [2009]	SVMs	P2P-TV	R: 91.3–99.6; FP: 0.3–8.7	testbed
Sena and Belzarena [2009]	SVMs	multiple protocols	A: 30–100	DPI
Palmieri and Fiore [2009]	behavioral signatures	HTTP, SMTP, DNS, POP, SSH, *eDonkey*	R: 45.8–89.4	DPI
Baldi et al. [2009]	service-based	multiple protocols	R: 81–93	client probe
Callado et al. [2010]	combination of methods	multiple protocols	P: 60–99; C: 90–100	DPI and controled traces

Table VI. Overview of studies for traffic classification that follow different approaches, including their ability to be applied to encrypted traffic and their performance, in terms of accuracy (A), precision (P), recall (R), sensitivity (Sens), specificity (Spec), completeness (C), false positives (FP), or false negatives (FN).

Appr.	Methods	Studies	Performance (%)	Encryption
Port based	port numbers identification	Saroiu et al. [2002a]	–	Apply
		Gerber et al. [2003]	–	Apply
		Fraleigh et al. [2003]	–	Apply
		Sen and Wang [2004]	–	Apply
DPI	payload strings	Sen et al. [2004]	FP: 0; FN: 0.00–9.90	Does not apply
		Moore and Papagiannaki [2005]	R: 99.99	Does not apply
		Guo and Qiu [2008]	FP: 0.00–11; FN: 0.33–0.5	Does not apply
		Cascarano et al. [2010a]	–	Does not apply
	automated signature extraction	Haffner et al. [2005]	P: 99.0–100; R: 86.6–99.9	Apply
		Park et al. [2008]	A: 97.39; FP: 0.39–10.40; FN: 0	Apply
		Finamore et al. [2009]	R: 99.6; FP: 0.34	Apply
		Mantia et al. [2010]	R: 97.62	Apply
	heuristics based on payload bytes	Ehlert and Petgang [2006]	–	Apply
	payload randomness	Dhamankar and King [2007]	–	Apply
	string matched using DFA	Smith et al. [2008]	–	Does not apply
Classification In The Dark	heuristics	Constantinou and Mavrommatis [2006]	FP: 7.6–42.4; FN: 8.5–12.7	Apply
		Perényi et al. [2006]	R: 97.19–99.14; FP: 0.3; FN: 0.8	Apply
		John and Tafvelin [2008]	R: 98	Apply
	social behavior	Karagiannis et al. [2005a]	P: 95–99; C: 80–90	Apply
		Iliofotou et al. [2008; 2009]	P: 95–96; R: 94–95	Apply
	statistical or behavioral signatures	Freire et al. [2008a; 2008b]	R: 90–100; FP: 2–5	Apply
		Dusi et al. [2008; 2009]	R: 82.45–100.00	Apply
		Lin et al. [2009]	R: 74–100; FP: 0–9; FN: 0–18	Apply
		Palmieri and Fiore [2009]	R: 45.8–89.4	Apply
	Naïve Bayes and neural networks	Moore and Zuev [2005]	P: 13.46–99.27; R: 93.73–96.29	Apply
		Schmidt and Soysal [2006]	FP: 22–28; FN: 16–26	Apply
		Auld et al. [2007]	A: 95–99	Apply
	clustering	Bernaille et al. [2006a; 2006b]	R: 36.0–100.0; FP: 0.0–3.6	Apply
		Erman et. al [2006a; 2006b; 2007a; 2007b]	A: 80–95; P: 71.42–100.00; R: 62.10–100.00	Apply
		Nguyen and Armitage [2006; 2008a]	P: 98.3–99.7; R: 96.0–98.9	Apply
		Bernaille and Teixeira [2007]	R: 81.20–100.00; FP: 0.00–2.30	Apply
	decision trees	Early et al. [2003]	R: 82–100	Apply
		Cao et al. [2008]	FP: 0.05–12.7; FN: 0–17.9	Apply
		Angevine and Zincir-Heywood [2008]	R: 94–99; FP: 1–26	Apply
		Branch et al. [2009]	P: 99; R: 98	Apply
	Markov chains and models	Wright et al. [2006]	R: 57.70–96.70; FP: 0.62–8.37	Apply
		Dainotti et al. [2008]	R: 90.23–100.00	Apply
	SVMs	Gonzá1ez-Castaño et al. [2006]	A: 78.7–90.2	Apply
		Este et al. [2008; 2009]	A: 92.53–98.83; R: 69.6–100.0	Apply
		Valenti et al. [2009]	R: 91.3–99.6; FP: 0.3–8.7	Apply
		Sena and Belzarena [2009]	A: 30–100	Apply
	other ML-based methods	Liu et al. [2007]	A: 78.5–99.8	Apply
		Raahemi et al. [2008a]	A: 78–92; Sens: 68–90; Spec: 85–96	Apply
		Huang et al. [2008]	R: 90.87–95.11	Apply
		Hu et al. [2008; 2009]	R: 90.0–98.0; FP: 0.2–5.0	Apply
	service identification	Baldi et al. [2009]	R: 81–93	Apply
Active Mechanisms	Active crawlers	Saroiu et al. [2002b; 2003]	–	Apply
		Ohzahata et al. [2005]	–	Apply
Combination of Approaches	DPI and heuristics	Karagiannis et al. [2004c]	R: 90–99; FP: 8–12	Apply
	product dists, Markov models, and substring graphs	Ma et al. [2006]	P: 0.0–100.0; R: 0.0–100.0	Does not apply
	DPI, heuristics, and ports	Szabó et al. [2007]	–	Does not apply
	Naïve Bayes and payload randomness	Bonfiglio et al. [2007]	FP: 0.00–2.40; FN: 2.96–29.9	Apply
	DPI and neural networks	Nogueira et al. [2007; 2009]	R: 90–99	Apply
	DPI and statistical analysis	Adami et al. [2009]	FP: 0.00–0.01; FN: 0.06–27.46	Apply
	combination of methods	Callado et al. [2010]	P: 60–99; R: 90–100	Apply

The literature review presented in this section shows a clear trend towards the use of classification *in the dark* methods. The majority of the articles published in the last few years proposed alternatives to DPI that can be used for encrypted or obfuscated traffic and can operate in real-time in high-speed networks. This tendency is driven by the growth of the networks throughput and need to have means to identify the nature of the traffic, and also by the increasingly common payload encryption.

The early methods for traffic classification *in the dark* were mostly based on behavior modeling, either by resorting to heuristics or by implementing more complex mechanisms. More recently, however, most studies are proposing classifiers based on statistical signatures or in multiple ML algorithms. Although the number of proposals based on ML is growing significantly, they seem to have reached a point were most of them use similar ML algorithms to process different features of the traffic and all of them present high accuracy. Hence, it is difficult to be sure if such proposals are evolving the state-of-the-art in traffic classification. Some of the recent articles are still proposing methods implemented to work off-line only, as they need to have access to the entire flows. Moreover, the methods for classification *in the dark* are growing in complexity, compromising one of their main motivations. In fact, Cascarano et al. [2009] compared the performance of a DPI classifier and an SVM-based method and concluded that they have similar computation cost. On the other hand, some recent studies are also proposing DPI methods that are able to classify encrypted traffic (see table III).

Therefore, more effort should be put on strategies to evaluate the true performance of the classifiers. This is not a simple task and it raises many challenges, as described in section 4.5. However, it is crucial, not to compare classifiers, but to have an accurate perception if the current proposals are really effective and how they can be improved. To the best of our knowledge, only three articles have addressed the subject of ground truth verification and proposed solutions [Szabó et al. 2008; Gringoli et al. 2009; Canini et al. 2009]. Moreover, a correct performance evaluation depends also on the datasets used for the validation. The classification challenges raised by several applications should be carefully analyzed and perhaps datasets of the traffic from many of them can be made available to be used in research studies.

Furthermore, there are several available tools, ready to use, for traffic classification using DPI; however, there are almost no applications implementing traffic characterization *in the dark* methods and that can be easily installed and experimented, on-line and off-line. Although this is not a clear research goal, it would be interesting to be able to effortlessly use some of the proposed methods in real-time experimental network environments and see how they could adapt to real scenarios.

6. CONCLUSIONS

The evolution of the services and applications running on the Internet has caused important changes in the properties of the traffic. Besides the increase of bandwidth consumption, other challenges have been raised for network managers. In order guarantee, the correct operation of networks, efficient mechanisms for traffic classification are required. Since port-based methods have lost their utility when the protocols started to use random port numbers, many studies proposed alternative mechanisms to classify traffic, either by deeply inspecting the traffic or using behavioral information.

This paper presents a survey on traffic classification that describes carefully the existing approaches. An extensive analysis of the literature was provided, pointing out the achievements and strengths of each study and its main goals. For the sake of understanding, it was also included an introduction to the subject of traffic measuring for the purpose of network monitoring.

The analysis of the literature bespeaks a clear interest of researchers, in the last years, in the traffic classification subject, motivated by the challenges created by new services and protocols, especially the ones based on P2P architecture. Furthermore, the evolution of the studies on this topic shows an increasing concern about the encryption of the traffic and its consequences for traffic management. The search for more accurate behavioral methods and DPI mechanisms capable of processing traffic in high-speed networks, together with the capability to classify encrypted traffic, seem to be strong trends for the future.

ACKNOWLEDGMENTS

The authors are thankful to all the anonymous reviewers for constructively criticizing this work.

REFERENCES

ADAMI, D., CALLEGARI, C., GIORDANO, S., PAGANO, M., AND PEPE, T. 2009. A real-time algorithm for Skype traffic detection and classification. In *Proceedings of the 9th International Conf. on Next Generation Wired / Wireless Networking (NEW2AN '09)* (St. Petersburg, Russia, Sept.). LNCS Series, vol. 5764. Springer-Verlag, Berlin Heidelberg, 168–179.

ALLMAN, M. AND PAXSON, V. 2007. Issues and etiquette concerning use of shared measurement data. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2007)* (San Diego, CA, USA, Oct.). ACM, New York, NY, USA, 135–140.

AMER, P. D. AND CASSEL, L. N. 1989. Management of sampled real-time network measurements. In *Proceedings of the 14th IEEE Conf. on Local Computer Networks (LCN '89)* (Mineapolis, MN, USA, Oct.). IEEE Press, New York, NY, USA, 62–68.

ANGEVINE, D. AND ZINCIR-HEYWOOD, A. N. 2008. A preliminary investigation of Skype traffic classification using a minimalist feature set. In *Proceedings of the 3rd International Conf. on Availability, Reliability and Security (ARES 08)* (Barcelona, Spain, Mar.). IEEE Computer Society Press, 1075–1079.

ANTONIADES, D., POLYCHRONAKIS, M., ANTONATOS, S., MARKATOS, E. P., UBIK, S., AND ØSLEBØ, A. 2006. Appmon: An application for accurate per application network traffic characterization. In *Proceedings of the IST Broadband Europe 2006 Conf.* (Geneva, Switzerland, Dec.).

APISDORF, J., CLAFFY, K. C., THOMPSON, K., AND WILDER, R. 1996. OC3MON: Flexible, affordable, high performance statistics collection. In *Proceedings of the 10th USENIX Systems Administration Conf. (LISA '96)* (Chicago, IL, USA, Sept.). USENIX Association, Berkeley, CA, USA, 97–112.

APPMON. 2010. Appmon description. `http://lobster.ics.forth.gr/~appmon/appmon_description.html`. Last access on March 24th, 2010.

ARLITT, M. AND WILLIAMSON, C. 2007. The extensive challenges of Internet application measurement. *IEEE Netw. 21,* 3, 41–46.

AULD, T., MOORE, A. W., AND GULL, S. F. 2007. Bayesian neural networks for Internet traffic classification. *IEEE Trans. Neural Netw. 18,* 1, 223–239.

AZZOUNA, N. B. AND GUILLEMIN, F. 2003. Analysis of ADSL traffic on an IP backbone link. In *Proceedings of the IEEE Global Communications Conf. (GLOBECOM '03)* (San Francisco, CA, USA, Dec.). Vol. 7. IEEE, 3742–3746.

BALDI, M., BALDINI, A., CASCARANO, N., AND RISSO, F. 2009. Service-based traffic classification: Principles and validation. In *Proceedings of the IEEE Sarnoff Symposium (SARNOFF '09)* (Princeton, NJ, USA, Mar./Apr.). IEEE Press, Piscataway, NJ, USA, 115–120.

BARTLETT, G., HEIDEMANN, J., AND PAPADOPOULOS, C. 2007a. Inherent behaviors for on-line detection of peer-to-peer file sharing. In *Proceedings of the IEEE Global Internet Symposium* (Anchorage, AK, USA, May). IEEE, 55–60.

BARTLETT, G., HEIDEMANN, J., AND PAPADOPOULOS, C. 2007b. Understanding passive and active service discovery. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2007)* (San Diego, CA, USA, Oct.). ACM, New York, NY, USA, 57–70.

BASHER, N., MAHANTI, A., MAHANTI, A., WILLIAMSON, C., AND ARLITT, M. 2008. A comparative analysis of web and peer-to-peer traffic. In *Proceedings of the 17th International Conf. on World Wide Web (WWW '08)* (Beijing, China, Apr.). ACM, New York, NY, USA, 287–296.

BERNAILLE, L. AND TEIXEIRA, R. 2007. Early recognition of encrypted applications. In *Proceedings of the Passive and Active Measurement Conf. (PAM 2007)* (Louvain-la-neuve, Belgium, Apr.). LNCS Series, vol. 4427. Springer-Verlag, Berlin Heidelberg, 165–175.

BERNAILLE, L., TEIXEIRA, R., AKODJENOU, I., SOULE, A., AND SALAMATIAN, K. 2006a. Traffic classification on the fly. *ACM SIGCOMM Comput. Commun. Rev. 36,* 2, 23–26.

BERNAILLE, L., TEIXEIRA, R., AND SALAMATIAN, K. 2006b. Early application identification. In *Proceedings of the 2nd Conf. on Future Networking Technologies (CoNEXT '06)* (Lisboa, Portugal, Dec.). ACM, 1–12.

BIN, L., ZHI-TANG, L., AND HAO, T. 2007. A methodology for P2P traffic measurement using application signature work-in-progress. In *Proceedings of the 2nd International Conf. on Scalable Information Systems (InfoScale '07)* (Suzhou, China, June). Vol. 304. ICST, Brussels, Belgium, 1–2.

BONFIGLIO, D., MELLIA, M., MEO, M., ROSSI, D., AND TOFANELLI, P. 2007. Revealing Skype traffic: When randomness plays with you. *ACM SIGCOMM Comput. Commun. Rev. 37,* 4, 37–48.

BRANCH, P. A., HEYDE, A., AND ARMITAGE, G. J. 2009. Rapid identification of Skype traffic flows. In *Proceedings of the 18th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '09)* (Williamsburg, VA, USA, June). ACM, New York, NY, USA, 91–96.

BRO. 2010. Bro intrusion detection system. `http://bro-ids.org`. Last access on March 24th, 2010.

CÁCERES, R., DUFFIELD, N., FELDMANN, A., FRIEDMANN, J. D., GREENBERG, A., GREER, R., JOHNSON, T., KALMANEK, C. R., KRISHNAMURTHY, B., LAVELLE, D., MISHRA, P. P., REXFORD, J., RAMAKRISHNAN, K. K., TRUE, F. D., AND VAN DER MERWE, J. E. 2000. Measurement and analysis of IP network usage and behavior. *IEEE Commun. Mag. 38,* 5, 144–151.

CALLADO, A., KAMIENSKI, C., SZABÓ, G., GERO, B. P., KELNER, J., FERNANDES, S., AND SADOK, D. 2009. A survey on Internet traffic identification. *IEEE Commun. Surveys Tuts. 11,* 3, 37–52.

CALLADO, A., KELNER, J., SADOK, D., KAMIENSKI, C. A., AND FERNANDES, S. 2010. Better network traffic identification through the independent combination of techniques. *J. Netw. Comput. Appl. 33,* 4, 433–446.

CANINI, M., LI, W., MOORE, A. W., AND BOLLA, R. 2009. GTVS: Boosting the collection of application traffic ground truth. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA '09)* (Aachen, Germany, May). Springer Verlag, Heidelberg, Germany, 54–63.

CAO, J., CHEN, A., WIDJAJA, I., AND ZHOU, N. 2008. Online identification of applications using statistical behavior analysis. In *Proceedings of the IEEE Global Telecommunications Conf. (GLOBECOM 2008)* (New Orleans, LA, USA, Nov./Dec.). IEEE, 1–6.

CARVALHO, D. A., PEREIRA, M., AND FREIRE, M. M. 2009a. Detection of peer-to-peer TV traffic through deep packet inspection. In *Acta da 9ª Conf. sobre Redes de Computadores* (Oeiras, Portugal, Oct.). INESC-ID and Instituto Superior Técnico, 6.

CARVALHO, D. A., PEREIRA, M., AND FREIRE, M. M. 2009b. Towards the detection of encrypted BitTorrent traffic through deep packet inspection. In *Proceedings of the International Conf. on Security Technology (SecTech 2009)* (Jeju Island, Korea, Dec.). Communications in Computer and Information Science Series, vol. 58. Springer-Verlag, Berlin Heidelberg, 265–272.

CASCARANO, N., CIMINIERA, L., AND RISSO, F. 2010a. Improving cost and accuracy of DPI traffic classifiers. In *Proceedings of the 25th ACM Symposium on Applied Computing (SAC 2010)* (Sierre, Switzerland, Mar.). ACM, New York, NY, USA, 641–646.

CASCARANO, N., ESTE, A., GRINGOLI, F., RISSO, F., AND SALGARELLI, L. 2009. An experimental evaluation of the computational cost of a DPI traffic classifier. In *Proceedings of the IEEE Global Communications Conf. (GLOBECOM 2009)* (Honolulu, HI, USA, Nov./Dec.). IEEE, 1–8.

CASCARANO, N., RISSO, F., ESTE, A., GRINGOLI, F., FINAMORE, A., AND MELLIA, M. 2010b. Comparing P2PTV traffic classifiers. In *Proceedings of the IEEE International Conf. on Communications (ICC 2010)* (Cape Town, South Africa, May). IEEE, 1–6.

CAVALLARO, L., LANZI, A., MAYER, L., AND MONGA, M. 2008. LISABETH: Automated content-based signature generator for zero-day polymorphic worms. In *Proceedings of the 4th International Workshop on Software Engineering for Secure Systems (SESS '08)* (Leipzig, Germany, May). ACM, New York, NY, USA, 41–48.

CHOI, K. AND CHOI, J. K. 2006. Pattern matching of packet payload for network traffic classification. In *Proceedings of the Joint International Conf. on Optical Internet and Next Generation Network (COIN-NGNCON 2006)* (Hyatt Regency Jehu, Korea, July). IEEE, 130–132.

CHOPRA, D., SCHULZRINNE, H., MAROCCO, E., AND IVOV, E. 2009. Peer-to-peer overlays for real-time communication: Security issues and solutions. *IEEE Commun. Surv. Tut. 11,* 1, 4–12.

CISCO NETFLOW. 2010. `http://www.cisco.com/web/go/netflow`. Last access on March 24th, 2010.

CLAFFY, K. C., BRAUN, H.-W., AND POLYZOS, G. C. 1995. A parameterizable methodology for Internet traffic flow profiling. *IEEE J. Sel. Areas Commun. 13,* 8, 1481–1494.

CLAFFY, K. C. AND MCCREARY, S. 1999. Internet measurement and data analysis: passive and active measurement. In *American Statistical Association* (Aug.).

CONSTANTINOU, F. AND MAVROMMATIS, P. 2006. Identifying known and unknown peer-to-peer traffic. In *Proceedings of 5th IEEE International Symposium on Network Computing and Applications (NCA '06)* (Cambridge, MA, USA, July). IEEE, 93–102.

COUTO, A., NOGUEIRA, A., SALVADOR, P., AND VALADAS, R. 2008. Identification of peer-to-peer applications' flow patterns. In *Proceedings of the Conf. on Next Generation Internet Networks (NGI 2008)* (Kraków, Poland, Apr.). IEEE, 292–299.

CROTTI, M., DUSI, M., GRINGOLI, F., AND SALGARELLI, L. 2007. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Comput. Commun. Rev. 37,* 1, 5–16.

CROTTI, M., GRINGOLI, F., PELOSATO, P., AND SALGARELLI, L. 2006. A statistical approach to IP-level classification of network traffic. In *Proceedings of the IEEE International Conf. on Communications (ICC '06)* (Istambul, Turkey, June). Vol. 1. IEEE, 170–176.

CROVELLA, M. AND KRISHNAMURTHY, B. 2006. *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley & Sons, Inc., New York, NY, USA.

DAINOTTI, A., DE DONATO, W., PESCAPÈ, A., AND ROSSI, P. S. 2008. Classification of network traffic via packet-level hidden markov models. In *Proceedings of the IEEE Global Telecommunications Conf. (GLOBECOM 2008)* (New Orleans, LA, USA, Nov./Dec.). IEEE, 1–5.

DAINOTTI, A., DE DONATO, W., PESCAPÉ, A., AND VENTRE, G. 2009. TIE: A community-oriented traffic classification platform. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA '09)* (Aachen, Germany, May). LNCS Series, vol. 5537. Springer-Verlag, Berlin Heidelberg, 64–74.

DEDINSKI, I., MEER, H. D., HAN, L., MATHY, L., PEZAROS, D. P., SVENTEK, J. S., AND XIAOYING, Z. 2005. Cross-layer peer-to-peer traffic identification and optimization based on active networking. In *Proceedings of the 7th Annual International Working Conf. on Active and Programmable Networks (IWAN 2005)* (Sophia Antipolis, France, Nov.). Springer-Verlag, Berlin Heidelberg, 13–27.

DEWES, C., WICHMANN, A., AND FELDMANN, A. 2003. An analysis of Internet chat systems. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2003)* (Miami Beach, FL, USA, Oct.). ACM, New York, NY, USA, 51–64.

DHAMANKAR, R. AND KING, R. 2007. Protocol identification via statistical analysis (PISA). *White Paper, Tipping Point*.

DUFFIELD, N., LUND, C., AND THORUP, M. 2005. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Trans. Netw. 13,* 5, 933–946.

DUFFIELD, N. G. 2004. Sampling for passive Internet measurement: A review. *Statistical Science 19,* 3, 472–498.

DUSI, M., CROTTI, M., GRINGOLI, F., AND SALGARELLI, L. 2008. Detection of encrypted tunnels across network boundaries. In *Proceedings of the IEEE International Conf. on Communications (ICC 2008)* (Beijing, China, May). IEEE, 1738–1744.

DUSI, M., CROTTI, M., GRINGOLI, F., AND SALGARELLI, L. 2009. Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting. *Comput. Netw. 53,* 1, 81–97.

EARLY, J. P., BRODLEY, C. E., AND ROSENBERG, C. 2003. Behavioral authentication of server flows. In *Proceedings of the 19th Annual Computer Security Applications Conf. (ACSAC '03)* (Las Vegas, NV, USA, Dec.). IEEE Computer Society, Los Alamitos, CA, USA, 46–55.

EHLERT, S. AND PETGANG, S. 2006. Analysis and signature of Skype VoIP session traffic. Tech. Rep. NGNI-SKYPE-06b, Fraunhofer FOKUS, Berlin, Germany. July.

ENDACE. 2011. Enterprise network monitoring tools – network security system – application performance monitoring. `http://www.endace.com`. Last access on July 28th, 2011.

ERMAN, J., ARLITT, M., AND MAHANTI, A. 2006a. Traffic classification using clustering algorithms. In *Proceedings of the ACM SIGCOMM Workshop on Mining Network Data (MineNet '06)* (Pisa, Italy, Sept.). ACM, New York, NY, USA, 281–286.

ERMAN, J., MAHANTI, A., AND ARLITT, M. 2006b. Internet traffic identification using machine learning. In *Proceedings of the IEEE Global Telecommunications Conf. (GLOBECOM 2006)* (San Francisco, CA, USA, Nov./Dec.). IEEE, 1–6.

ERMAN, J., MAHANTI, A., ARLITT, M., COHEN, I., AND WILLIAMSON, C. 2007a. Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation 64,* 9-12, 1194–1213.

ERMAN, J., MAHANTI, A., ARLITT, M., AND WILLIAMSON, C. 2007b. Identifying and discriminating between web and peer-to-peer traffic in the network core. In *Proceedings of the 16th International Conf. on World Wide Web (WWW 2007)* (Banff, Alberta, Canada, May). ACM Press, New York, NY, USA, 883–892.

ESTE, A., GARGIULO, F., GRINGOLI, F., SALGARELLI, L., AND SANSONE, C. 2008. Pattern recognition approaches for classifying IP flows. In *Proceedings of the Joint IAPR International Workshop on Struc-*

tural, Syntactic, and Statistical Pattern Recognition (SSPR & SPR '08) (Orlando, FL, USA, Dec.). LNCS Series, vol. 5342. Springer-Verlag, Berlin Heidelberg, 885–895.

ESTE, A., GRINGOLI, F., AND SALGARELLI, L. 2009. Support vector machines for TCP traffic classification. *Comput. Netw. 53,* 14, 2476–2490.

ETTERCAP. 2010. `http://ettercap.sourceforge.net`. Last access on March 24th, 2010.

FINAMORE, A., MELLIA, M., MEO, M., AND ROSSI, D. 2009. KISS: Stochastic packet inspection. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA '09)* (Aachen, Germany, May). LNCS Series, vol. 5537. Springer-Verlag, Berlin Heidelberg, 117–125.

FRALEIGH, C., MOON, S., LYLES, B., COTTON, C., KHAN, M., MOLL, D., ROCKELL, R., SEELY, T., AND DIOT, C. 2003. Packet-level traffic measurements from the Sprint IP backbone. *IEEE Netw. 17,* 6, 6–16.

FREIRE, E. P., ZIVIANI, A., AND SALLES, R. M. 2008a. Detecting Skype flows in web traffic. In *Proceedings of the IEEE Network Operations and Management Symposium (NOMS 2008)* (Salvador da Bahia, Brazil, Apr.). IEEE, 89–96.

FREIRE, E. P., ZIVIANI, A., AND SALLES, R. M. 2008b. Detecting VoIP calls hidden in web traffic. *IEEE Trans. Netw. Service Manag. 5,* 4, 204–214.

FREIRE, M. M., CARVALHO, D. A., AND PEREIRA, M. 2009. Detection of encrypted traffic in eDonkey network through application signatures. In *Proceedings of the 1st International Conf. on Advances in P2P Systems (AP2PS 2009)* (Sliema, Malta, Oct.). IEEE Computer Society Press, Los Alamitos, CA, USA, 174–179.

GERBER, A., HOULE, J., NGUYEN, H., ROUGHAN, M., AND SEN, S. 2003. P2P, the gorilla in the cable. In *Proceedings of the National Cable & Telecommunications Association (NCTA)* (Chicago, IL, USA, June). 8–11.

GOMES, J. V. P., INÁCIO, P. R. M., FREIRE, M. M., PEREIRA, M., AND MONTEIRO, P. P. 2008. Analysis of peer-to-peer traffic using a behavioural method based on entropy. In *Proceedings of the 27th IEEE International Performance Computing and Communications Conf. (IPCCC 2008)* (Austin, TX, USA, Dec.). IEEE Computer Society Press, Los Alamitos, CA, USA, 201–208.

GONZÁ1EZ-CASTAÑO, F. J., RODRÍGUEZ-HERNÁNDEZ, P. S., MARTÍNEZ-ÁLVAREZ, R. P., GÓMEZ, A., LÓPEZ-CABIDO, I., AND VILLASUSO-BARREIRO, J. 2006. Support vector machine detection of peer-to-peer traffic. In *Proceedings of IEEE International Conf. on Computational Intelligence for Measurement Systems and Applications (CIMSA 2006)* (La Coruña, Spain, July). IEEE, 103–108.

GRINGOLI, F., SALGARELLI, L., DUSI, M., CASCARANO, N., RISSO, F., AND CLAFFY, K. C. 2009. GT: Picking up the truth from the ground for Internet traffic. *ACM SIGCOMM Comput. Commun. Rev. 39,* 5, 13–18.

GUO, Z. AND QIU, Z. 2008. Identification peer-to-peer traffic for high speed networks using packet sampling and application signatures. In *Proceedings of the 9th International Conf. on Signal Processing (ICSP 2008)* (Beijing, China, Oct.). IEEE, 2013–2019.

HAFFNER, P., SEN, S., SPATSCHECK, O., AND WANG, D. 2005. ACAS: Automated construction of application signatures. In *Proceedings of the ACM SIGCOMM Workshop on Mining Network Data (MineNet '05)* (Philadelphia, PA, USA, Aug.). ACM, New York, NY, USA, 197–202.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl. 11,* 1, 10–18.

HU, Y., CHIU, D.-M., AND LUI, J. C. S. 2008. Application identification based on network behavioral profiles. In *Proceedings of the 16th International Workshop on Quality of Service (IWQoS 2008)* (Enschede, The Netherlands, June). IEEE, 219–228.

HU, Y., CHIU, D.-M., AND LUI, J. C. S. 2009. Profiling and identification of P2P traffic. *Comput. Netw. 53,* 6, 849–863.

HUANG, N.-F., JAI, G.-Y., AND CHAO, H.-C. 2008. Early identifying application traffic with application characteristics. In *Proceedings of the IEEE International Conf. on Communications (ICC 2008)* (Beijing, China, May). IEEE, 5788–5792.

IANA. 2011. Port numbers. Last access on July 28th, 2011.

IETF. 2008. Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. *RFC 5101*.

ILIOFOTOU, M., KIM, H.-C., FALOUTSOS, M., MITZENMACHER, M., PAPPU, P., AND VARGHESE, G. 2009. Graph-based P2P traffic classification at the Internet backbone. In *Proceedings of the 28th IEEE International Conf. on Computer Communications Workshops (INFOCOM '09)* (Rio de Janeiro, Brazil, Apr.). IEEE Press, Piscataway, NJ, USA, 37–42.

ILIOFOTOU, M., PAPPU, P., FALOUTSOS, M., MITZENMACHER, M., SINGH, S., AND VARGHESE, G. 2007. Network monitoring using traffic dispersion graphs (TDGs). In *Proceedings of the ACM SIGCOMM*

Internet Measurement Conf. (IMC 2007) (San Diego, CA, USA, Oct.). ACM, New York, NY, USA, 315–320.

ILIOFOTOU, M., PAPPU, P., FALOUTSOS, M., MITZENMACHER, M., VARGHESE, G., AND KIM, H. 2008. Graption: Automated detection of P2P applications using traffic dispersion graphs (TDGs). Tech. Rep. UCR-CS-2008-06080. June.

INOUE, H., JANSENS, D., HIJAZI, A., AND SOMAYAJI, A. 2007. NetADHICT: A tool for understanding network traffic. In *Proceedings of the 21st Large Installation System Administration Conf. (LISA '07)* (Dallas, TX, USA, Nov.). USENIX Association, 39–47.

IPOQUE. 2011. Bandwidth management with deep packet inspection. `http://www.ipoque.com`. Last access on July 28th, 2011.

JAIN, R. AND ROUTHIER, S. A. 1986. Packet trains–measurements and a new model for computer network traffic. *IEEE J. Sel. Areas Commun. 4,* 6, 986–995.

JOHN, W. AND TAFVELIN, S. 2008. Heuristics to classify Internet backbone traffic based on connection patterns. In *Proceedings of the International Conf. on Information Networking (ICOIN 2008)* (Busan, Korea, Jan.). IEEE, 1–5.

JOHNSON, M. E., MCGUIRE, D., AND WILLEY, N. D. 2008. The evolution of the peer-to-peer file sharing industry and the security risks for users. In *Proceedings of the Proceedings of the 41st Hawaii International Conf. on System Sciences (HICSS 2008)* (Waikoloa, HI, USA, Jan.). IEEE Computer Society, Washington, DC, USA.

JOHNSON, M. E., MCGUIRE, D., AND WILLEY, N. D. 2009. Why file sharing networks are dangerous? *Commun. ACM 52,* 2, 134–138.

JURGA, R. E. AND HULBÓJ, M. M. 2007. Packet sampling for network monitoring. Tech. rep., CERN — HP Procurve openlab project. Dec.

KARAGIANNIS, T., BROIDO, A., BROWNLEE, N., CLAFFY, K., AND FALOUTSOS, M. 2004a. File-sharing in the Internet: A characterization of P2P traffic in the backbone. Tech. rep.

KARAGIANNIS, T., BROIDO, A., BROWNLEE, N., CLAFFY, K. C., AND FALOUTSOS, M. 2004b. Is P2P dying or just hiding? In *Proceedings of the IEEE Global Telecommunications Conf. (GLOBECOM '04)* (Dallas, TX, USA, Nov./Dec.). Vol. 3. IEEE Computer Society Press, Piscataway, NJ, USA, 1532–1538.

KARAGIANNIS, T., FALOUTSOS, A. B. M., AND CLAFFY, K. C. 2004c. Transport layer identification of P2P traffic. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2004)* (Taormina, Sicily, Italy, Oct.). ACM, New York, NY, USA, 121–134.

KARAGIANNIS, T., PAPAGIANNAKI, K., AND FALOUTSOS, M. 2005a. BLINC: Multilevel traffic classification in the dark. In *Proceedings of the ACM SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Philadelphia, PA, USA, Aug.). Vol. 35. ACM, New York, NY, USA, 229–240.

KARAGIANNIS, T., RODRIGUEZ, P., AND PAPAGIANNAKI, K. 2005b. Should Internet service providers fear peer-assisted content distribution? In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2005)* (Berkeley, CA, USA, Oct.). USENIX Association, Berkeley, CA, USA, 63–76.

KIM, H., CLAFFY, K. C., FOMENKOV, M., BARMAN, D., FALOUTSOS, M., AND LEE, K. 2008. Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proceedings of the ACM International Conf. on emerging Networking EXperiments and Technologies (CoNEXT '08)* (Madrid, Spain, Dec.). ACM, New York, NY, USA, 1–12.

KIM, H.-C., FOMENKOV, M., BROWNLEE, N., CLAFFY, K. C., BARMAN, D., AND FALOUTSOS, M. 2007. Comparison of Internet traffic classification tools. In *Workshop on Application Classification and Identification (WACI)* (Boston, MA, USA, Oct.).

KIND, A., DIMITROPOULOS, X., DENAZIS, S., AND CLAISE, B. 2008. Advanced network monitoring brings life to the awareness plane. *IEEE Commun. Mag. 46,* 10, 140–146.

KRISHNAMURTHY, B. AND WANG, J. 2002. Traffic classification for application specific peering. In *Proceedings of the 2nd ACM SIGCOMM Internet Measurement Workshop (IMW '02)* (Marseille, France, Nov.). ACM, New York, NY, USA, 179–180.

KUMAR, S., DHARMAPURIKAR, S., YU, F., CROWLEY, P., AND TURNER, J. 2006. Algorithms to accelerate multiple regular expressions matching for deep packet inspection. *ACM SIGCOMM Comput. Commun. Rev. 36,* 4, 339–350.

L7-FILTER. 2010. L7-filter, application layer packet classifier for Linux. `http://l7-filter.sourceforge.net`. Last access on March 24th, 2010.

L7-NETPDLCLASSIFIER. 2010. Tools for L2-L7 traffic classification. `http://netgroup.polito.it/research-projects/l7-traffic-classification/`. Last access on March 24th, 2010.

LAKHINA, A., CROVELLA, M., AND DIOT, C. 2005. Mining anomalies using traffic feature distributions. *ACM SIGCOMM Comput. Commun. Rev. 35,* 4, 217–228.

LAWTON, G. 2004. Is peer-to-peer secure enough for corporate use? *IEEE Comput. 37,* 1, 22–25.

LEIBOWITZ, N., BERGMAN, A., BEN-SHAUL, R., AND SHAVIT, A. 2002. Are file swapping networks cacheable? Characterizing P2P traffic. In *Proceedings of the 7th International Workshop on Web Content Caching and Distribution (WCW)* (Boulder, CO, USA, Aug.).

LI, T., GUAN, Z., AND WU, X. 2007. Modeling and analyzing the spread of active worms based on P2P systems. *Comput. & Security 26,* 3, 213–218.

LI, W., CANINI, M., MOORE, A. W., AND BOLLA, R. 2009. Efficient application identification and the temporal and spatial stability of classification schema. *Comput. Netw. 53,* 6, 790–809.

LIN, Y.-D., LU, C.-N., LAI, Y.-C., PENG, W.-H., AND LIN, P.-C. 2009. Application classification using packet size distribution and port association. *J. Netw. Comput. Appl. 32,* 5, 1023–1030.

LIU, H., FENG, W., HUANG, Y., AND LI, X. 2007. A peer-to-peer traffic identification method using machine learning. In *Proceedings of the International Conf. on Networking, Architecture, and Storage (NAS 2007)* (Guilin, China, July). IEEE, 155–160.

MA, J., LEVCHENKO, K., KREIBICH, C., SAVAGE, S., AND VOELKER, G. M. 2006. Unexpected means of protocol inference. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2006)* (Rio de Janeiro, Brazil, Oct.). ACM, New York, NY, USA, 313–326.

MADHUKAR, A. AND WILLIAMSON, C. 2006. A longitudinal study of P2P traffic classification. In *Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2006)* (Monterey, CA, USA, Sept.). IEEE Computer Society, Washington, DC, USA, 179–188.

MAKHOUL, J., KUBALA, F., SCHWARTZ, R., AND WEISCHEDEL, R. 1999. Performance measures for information extraction. In *Proceedings of the DARPA Broadcast News Workshop* (Herndon, VA, USA, Feb.). 249–252.

MANTIA, G. L., ROSSI, D., FINAMORE, A., MELLIA, M., AND MEO, M. 2010. Stochastic packet inspection for TCP traffic. In *Proceedings of the IEEE International Conf. on Communications (ICC 2010)* (Cape Town, South Africa, May). IEEE, 1–6.

MAPI. 2010. MAPI, monitoring API. `http://mapi.uninett.no`. Last access on March 24th, 2010.

MCGREGOR, A., HALL, M., LORIER, P., AND BRUNSKILL, J. 2004. Flow clustering using machine learning techniques. In *Proceedings of the Passive and Active Measurement Workshop (PAM 2004)* (Antibes Juanles-Pins, France, Apr.). LNCS Series, vol. 3015. Springer-Verlag, Berlin Heidelberg, 205–214.

MCGREGOR, T. 2002. Quality in measurement: beyond the deployment barrier. In *Proceedings of the Symposium on Applications and the Internet Workshops (SAINT-W '02)* (Nara, Japan, Feb.). IEEE Computer Society, Washington, DC, USA, 66–73.

MOORE, A. W. AND PAPAGIANNAKI, K. 2005. Toward the accurate identification of network applications. In *Proceedings of the Passive and Active Measurement Conf. (PAM 2005)* (Boston, MA, USA, Mar.). LNCS Series, vol. 3431. Springer-Verlag, Berlin Heidelberg, 41–54.

MOORE, A. W. AND ZUEV, D. 2005. Internet traffic classification using bayesian analysis techniques. *ACM SIGMETRICS Performance Evaluation Rev. 33,* 1, 50–60.

MOORE, A. W., ZUEV, D., AND CROGAN, M. L. 2005. Discriminators for use in flow-based classification. Tech. Rep. RR-05-13, Intel Research, Cambridge, UK. Aug.

MOORE, D., KEYS, K., KOGA, R., LAGACHE, E., AND CLAFFY, K. C. 2001. The CoralReef software suite as a tool for system and network administrators. In *Proceedings of the 15th USENIX System Administration Conf. (LISA '01)* (San Diego, CA, USA, Dec.). USENIX Association, Berkeley, CA, USA, 133–144.

MU, J., SEZER, S., DOUGLAS, G., BURNS, D., GARCIA, E., HUTTON, M., AND CACKOVIC, K. 2007. Accelerating pattern matching for DPI. In *Proceedings of the IEEE International Symposium on System-on-Chip (SOC 2007)* (Tampere, Finland, Sept.). IEEE, 83–86.

MURRAY, M. AND CLAFFY, K. C. 2001. Measuring the immeasurable: Global Internet measurement infrastructure. In *Proceedings of the Passive and Active Measurement Workshop (PAM 2001)* (Amsterdam, The Netherlands, Apr.). 159–167.

NAPATECH. 2011. Intelligent real-time network analysis. `http://www.napatech.com`. Last access on July 28th, 2011.

NETADHICT. 2010. `http://www.ccsl.carleton.ca/software/netadhict/`. Last access on March 24th, 2010.

NETBEE. 2010. The NetBee library. `http://www.nbee.org`. Last access on March 24th, 2010.

NETPDL. 2010. `http://www.nbee.org/netpdl`. Last access on March 24th, 2010.

NGUYEN, T. T. T. AND ARMITAGE, G. 2006. Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world IP networks. In *Proceedings of the IEEE Conf. on Local Computer Networks (LCN 2006)* (Tampa, FL, USA, Nov.). IEEE, 369–376.

NGUYEN, T. T. T. AND ARMITAGE, G. 2008a. Clustering to assist supervised machine learning for real-time IP traffic classification. In *Proceedings of the IEEE International Conf. on Communications (ICC 2008)* (Beijing, China, May). IEEE, 5857–5862.

NGUYEN, T. T. T. AND ARMITAGE, G. 2008b. A survey of techniques for Internet traffic classification using machine learning. *IEEE Commun. Surveys Tuts. 10,* 4, 56–76.

NLANR. 2010. NLANR/MNA home page. `http://www.nlanr.net`. Last access on March 24th, 2010.

NOGUEIRA, A., SALVADOR, P., COUTO, A., AND VALADAS, R. 2009. Towards the on-line identification of peer-to-peer flow patterns. *J. Netw. 4,* 2, 108–118.

NOGUEIRA, A., SALVADOR, P., AND VALADAS, R. 2007. A framework for detecting internet applications. In *Proceedings of the International Conf. on Information Networking (ICOIN 2007)* (Estoril, Portugal, Jan.). Springer-Verlag, Berlin Heidelberg, 455–464.

OHM, P., SICKER, D. C., AND GRUNWALD, D. 2007. Legal issues surrounding monitoring during network research. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2007)* (San Diego, CA, USA, Oct.). ACM, New York, NY, USA, 141–148.

OHZAHATA, S., HAGIWARA, Y., TERADA, M., AND KAWASHIMA, K. 2005. A traffic identification method and evaluations for a pure P2P application. In *Proceedings of the Passive and Active Measurement Conf. (PAM 2005)* (Boston, MA, USA, Mar.2005). LNCS Series, vol. 3431. Springer-Verlag, Berlin Heidelberg, 55–68.

OLSON, D. L. AND DELEN, D. 2008. *Advanced Data Mining Techniques* 1 Ed. Springer.

OPENDPI. 2010. OpenDPI - the open source deep packet inspection engine. `http://www.opendpi.org`. Last access on March 24th, 2010.

PALMIERI, F. AND FIORE, U. 2009. A nonlinear, recurrence-based approach to traffic classification. *Comput. Netw. 53,* 6, 761–773.

PARK, B.-C., WON, Y. J., KIM, M.-S., AND HONG, J. W. 2008. Towards automated application signature generation for traffic identification. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2008)* (Salvador da Bahia, Brazil, Apr.). IEEE, 160–167.

PAXSON, V. 2004. Strategies for sound Internet measurement. In *Proceedings of the ACM SIGCOMM Internet Measurement Conf. (IMC 2004)* (Taormina, Sicily, Italy, Oct.). ACM, New York, NY, USA, 263–271.

PERÉNYI, M., DANG, T. D., GEFFERTH, A., AND MOLNÁR, S. 2006. Identification and analysis of peer-to-peer traffic. *J. Commun. 1,* 7, 36–46.

PLONKA, D. 2000. FlowScan: A network traffic flow reporting and visualization tool. In *Proceedings of the 14th USENIX System Administration Conf. (LISA '00)* (New Orleans, LA, USA, Dec.). USENIX Association, Berkeley, CA, USA, 305–317.

RAAHEMI, B., KOUZNETSOV, A., HAYAJNEH, A., AND RABINOVITCH, P. 2008a. Classification of peer-to-peer traffic using incremental neural networks (fuzzy ARTMAP). In *Proceedings of the Canadian Conf. on Electrical and Computer Engineering (CCECE 2008)* (Niagara Falls, ON, Canada, May). IEEE, 719–724.

RAAHEMI, B., ZHONG, W., AND LIU, J. 2008b. Peer-to-peer traffic identification by mining IP layer data streams using concept-adapting very fast decision tree. In *Proceedings of the 20th IEEE International Conf. on Tools with Artificial Intelligence (ICTAI '08)* (Dayton, OH, USA, Nov.). Vol. 1. IEEE, 525–532.

RANJAN, S., SHAH, S., NUCCI, A., MUNAFÒ, M., CRUZ, R., AND MUTHUKRISHNAN, S. 2007. DoWitcher: Effective worm detection and containment in the internet core. In *Proceedings of the 26th IEEE International Conf. on Computer Communications (INFOCOM 2007)* (Anchorage, AK, USA, May). IEEE, 2541–2545.

RISSO, F. AND BALDI, M. 2006. NetPDL: an extensible XML-based language for packet header description. *Comput. Netw. 50,* 5, 688–706.

RISSO, F., BALDI, M., MORANDI, O., BALDINI, A., AND MONCLUS, P. 2008. Lightweight, payload-based traffic classification: An experimental evaluation. In *Proceedings of the IEEE International Conf. on Communications (ICC '08)* (Beijing, China, May). IEEE, 5869–5875.

ROMIG, S., FULLMER, M., AND LUMAN, R. 2000. The OSU flow-tools package and CISCO NetFlow logs. In *Proceedings of the 14th USENIX System Administration Conf. (LISA '00)* (New Orleans, LA, USA, Dec.). USENIX Association, Berkeley, CA, USA, 291–303.

SALGARELLI, L., GRINGOLI, F., AND KARAGIANNIS, T. 2007. Comparing traffic classifiers. *ACM SIGCOMM Comput. Commun. Rev. 37,* 3, 65–68.

SAROIU, S., GUMMADI, K. P., DUNN, R. J., GRIBBLE, S. D., AND LEVY, H. M. 2002a. An analysis of Internet content delivery systems. In *Proceedings of the 5th Symposium on Operating Systems Design andImplementation (OSDI '02)* (Boston, MA, USA, Dec.). Vol. 36. ACM, New York, NY, USA, 315–327.

SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. D. 2002b. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the Multimedia Computing and Networking (MMCN '02)* (San Jose, CA, USA, Jan.). ACM, New York, NY, USA.

SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. D. 2003. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Syst. J. 9,* 2, 170–184.

SCHMIDT, Ş. E. G. AND SOYSAL, M. 2006. An intrusion detection based approach for the scalable detection of P2P traffic in the national academic backbone network. In *Proceedings of the International Symposium on Computer Networks (ISCN 2006)* (June). IEEE, Istanbul, Turkey, 128–133.

SCHULZE, H. AND MOCHALSKI, K. 2007. Internet study 2007. Tech. rep., ipoque.

SCHULZE, H. AND MOCHALSKI, K. 2009. Internet study 2008/2009. Tech. rep., ipoque.

SEEDORF, J. 2006. Security challenges for peer-to-peer SIP. *IEEE Netw. 20,* 5, 38–45.

SEN, S., SPATSCHECK, O., AND WANG, D. 2004. Accurate, scalable in-network identification of P2P traffic using application signatures. In *Proceedings of the 13th International Conf. on World Wide Web (WWW '04)* (New York, NY, USA, May). ACM, New York, NY, USA, 512–521.

SEN, S. AND WANG, J. 2004. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Trans. Netw. 12,* 2, 219–232.

SENA, G. G. AND BELZARENA, P. 2009. Early traffic classification using support vector machines. In *Proceedings of the 5th International Latin American Networking Conf. (LANC '09)* (Pelotas, Brazil, Sept.). ACM, New York, NY, USA, 60–66.

SINGH, S., ESTAN, C., VARGHESE, G., AND SAVAGE, S. 2004. Automated worm fingerprinting. In *Proceedings of the 6th Symposium on Operating Systems Design & Implementation (OSDI '04)* (San Francisco, CA, USA, Dec.). USENIX Association, Berkeley, CA, USA, 45–60.

SMITH, F. D., CAMPOS, F. H., JEFFAY, K., AND OTT, D. 2001. What TCP/IP protocol headers can tell us about the Web. *ACM SIGMETRICS Performance Evaluation Rev. 29,* 1, 245–256.

SMITH, R., ESTAN, C., JHA, S., AND KONG, S. 2008. Deflating the big bang: Fast and scalable deep packet inspection with extended finite automata. *ACM SIGCOMM Comput. Commun. Rev. 38,* 4, 207–218.

SNORT. 2010. `http://www.snort.org`. Last access on March 24th, 2010.

SOEWITO, B., MAHAJAN, A., WENG, N., AND WANG, H. 2009. High-speed string matching for network intrusion detection. *Int. J. Commun. Netw. Distrib. Syst. 3,* 4, 319–339.

SOYSAL, M. AND SCHMIDT, E. G. 2007. An accurate evaluation of machine learning algorithms for flow-based P2P traffic detection. In *Proceedings of the 22nd International Symposium on Computer and Information Sciences (ISCIS 2007)* (Ankara, Turkey, Nov.). IEEE.

SPEROTTO, A., SADRE, R., VAN VLIET, F., AND PRAS, A. 2009. A labeled data set for flow-based intrusion detection. In *Proceedings of the 9th IEEE International Workshop on IP Operations and Management (IPOM 2009)* (Venice, Italy, Oct.). LNCS Series, vol. 5843. Springer-Verlag, Berlin Heidelberg, 39–50.

SPOGNARDI, A., LUCARELLI, A., AND PIETRO, R. D. 2005. A methodology for P2P file-sharing traffic detection. In *Proceedings of the 2nd International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P '05)* (La Jolla, CA, USA, July). IEEE Computer Society, Washington, DC, USA, 52–61.

STEFANOWSKI, J. AND WILK, S. 2009. Extending rule-based classifiers to improve recognition of imbalanced classes. In *Advances in Data Management*, Z. W. Ras and A. Dardzinska, Eds. SCI Series, vol. 223. Springer-Verlag Berlin Heidelberg, 131–154.

STRAYER, T., ARMITAGE, G., ALLMAN, M., MOORE, A. W., JIN, S., AND BELLOVIN, S. 2008. IMRG workshop on application classification and identification report. *ACM SIGCOMM Comput. Commun. Rev. 38,* 3, 87–90.

SZABÓ, G., ORINCSAY, D., MALOMSOKY, S., AND SZABÓ, I. 2008. On the validation of traffic classification algorithms. In *Proceedings of the Passive and Active Measurement Conf. (PAM 2008)* (Cleveland, OH, USA, Apr.). LNCS Series, vol. 4979. Springer-Verlag, Berlin Heidelberg, 72–81.

SZABÓ, G., SZABÓ, I., AND ORINCSAY, D. 2007. Accurate traffic classification. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)* (Helsinki, Finland, June). IEEE, 1–8.

TCPDUMP. 2011. TCPDUMP/LIBPCAP public repository. `http://www.tcpdump.org`. Last access on July 24th, 2011.

TIE. 2010. TIE, traffic identification engine. `http://tie.comics.unina.it`. Last access on March 24th, 2010.

TURKETT, W. H., KARODE, A. V., AND FULP, E. W. 2008. In-the-dark network traffic classification using support vector machines. In *Proceedings of the 20th National Conf. on Innovative Applications of Artificial Intelligence (IAAI '08)* (Chicago, IL, USA, July). AAAI Press, 1745–1750.

VALENTI, S., ROSSI, D., MEO, M., MELLIA, M., AND BERMOLEN, P. 2009. Accurate, fine-grained classification of P2P-TV applications by simply counting packets. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA '09)* (Aachen, Germany, May). LNCS Series, vol. 5537. Springer-Verlag, Berlin, Heidelberg, Berlin Heidelberg, 84–92.

WANG, Y. 2008. *Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection*. Premier Reference Source. Information Science Reference.

WANG, Y.-H., GAU, V., BOSAW, T., HWANG, J.-N., LIPPMAN, A., LIEBENNAN, D., AND WU, I.-C. 2008. Generalization performance analysis of flow-based peer-to-peer traffic identification. In *Proceedings of the IEEE Workshop on Machine Learning for Signal Processing (MLSP 2008)* (Cancún, Mexico, Oct.). IEEE, 267–272.

WEISS, G. M. 2004. Mining with rarity: A unifying framework. *ACM SIGKDD Explor. Newsl. 6,* 1, 7–19.

WILDPACKETS. 2011. WildPackets: Network analyzer, voip monitoring, protocol analysis. `http://www.wildpackets.com`. Last access on July 28th, 2011.

WILLIAMS, N., ZANDER, S., AND ARMITAGE, G. 2006. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Comput. Commun. Rev. 36,* 5, 5–16.

WILLIAMSON, C. 2001. Internet traffic measurement. *IEEE Internet Comput. 5,* 6, 70–74.

WINDUMP. 2011. tcpdump for Windows using WinPcap. `http://www.winpcap.org/windump/`. Last access on July 24th, 2011.

WINPCAP. 2011. The industry-standard windows packet capture library. `http://www.winpcap.org`. Last access on July 24th, 2011.

WIRESHARK. 2010. Wireshark, go deep. `http://www.wireshark.org`. Last access on March 24th, 2010.

WRIGHT, C. V., MONROSE, F., AND MASSON, G. M. 2006. On inferring application protocol behaviors in encrypted network traffic. *J. Mach. Learning Research 7*, 2745–2769.

XU, K., LIU, J., AND WANG, H. 2008. Tod-cache: Peer-to-peer traffic management and optimization using combined caching and redirection. In *Proceedings of the IEEE Global Telecommunications Conf. (GLOBECOM 2008)* (New Orleans, LA, USA, Nov./Dec.). IEEE, 1–5.

XUSHENG, Z. AND ZHIMING, W. 2009. Application of markov chain in IP traffic classification. In *Proceedings of the International Conf. on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)* (Wuhan, China, Apr.). Vol. 2. IEEE Computer Society, 688–691.

YEGNESWARAN, V., GIFFIN, J. T., BARFORD, P., AND JHA, S. 2005. An architecture for generating semantics-aware signatures. In *Proceedings of the 14th USENIX Security Symposium (SSYM '05)* (Baltimore, MD, USA, July/Aug.). USENIX Association, Berkeley, CA, USA, 97–112.

YU, F. 2006. High speed deep packet inspection with hardware support. Ph.D. thesis, EECS Department, University of California, Berkeley, CA, USA.

ZANDER, S., NGUYEN, T., AND ARMITAGE, G. 2005a. Automated traffic classification and application identification using machine learning. In *Proceedings of the IEEE Conf. on Local Computer Networks (LCN 2005)* (Sydney, Australia, Nov.). IEEE, 250–257.

ZANDER, S., NGUYEN, T., AND ARMITAGE, G. 2005b. Self-learning IP traffic classification based on statistical flow characteristics. In *Proceedings of the Passive and Active Measurement Conf. (PAM 2005)* (Boston, MA, USA, Mar.). LNCS Series, vol. 3431. Springer-Verlag, Berlin Heidelberg, 325–328.

ZHOU, L., ZHANG, L., MCSHERRY, F., IMMORLICA, N., COSTA, M., AND CHIEN, S. 2005. A first look at peer-to-peer worms: Threats and defenses. In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS 2005)* (Ithaca, NY, USA, Feb.). LNCS Series, vol. 3640. Springer, Berlin Heidelberg, 24–35.

ZUEV, D. AND MOORE, A. W. 2005. Traffic classification using a statistical approach. In *Proceedings of the Passive and Active Measurement Conf. (PAM 2005)* (Boston, MA, USA, Mar.). LNCS Series, vol. 3431. Springer-Verlag, Berlin Heidelberg, 321–324.

Received Month Year; revised Month Year; accepted Month Year

Chapter 3

Source Traffic Analysis

This chapter consists of the following article:

Source Traffic Analysis

João V. P. Gomes, Pedro. R. M. Inácio, Blanka Lakic, Mário M. Freire, Henrique J. A. da Silva, and Paulo P. Monteiro

ACM Transactions on Multimedia Computing, Communications and Applications, 6(3): Article 21, 1–23, 2010.

DOI: 10.1145/1823746.1823755

According to 2010 Journal Citation Reports published by Thomson Reuters in 2011, this journal scored ISI journal performance metrics as follows:

ISI Impact Factor (2010): 1.425

ISI Article Influence Score (2010): 0.894

Journal Ranking (2010): 51/128 (Computer Science, Information Systems)

Journal Ranking (2010): 26/99 (Computer Science, Software Engineering)

Journal Ranking (2010): 31/97 (Computer Science, Theory & Methods)

Source Traffic Analysis

JOÃO V. P. GOMES, PEDRO R. M. INÁCIO, Nokia Siemens Networks Portugal and University of Beira Interior
BRANKA LAKIC, University of Coimbra and Institute of Telecommunications
MÁRIO M. FREIRE, University of Beira Interior and Institute of Telecommunications
HENRIQUE J. A. DA SILVA, University of Coimbra and Institute of Telecommunications
PAULO P. MONTEIRO, Nokia Siemens Networks Portugal and Institute of Telecommunications

Traffic modeling and simulation plays an important role in the area of Network Monitoring and Analysis, for it provides practitioners with efficient tools to evaluate the performance of networks and of their elements. This article focus on the traffic generated by a single source, providing an overview of what was done in the field and studying the statistical properties of the traffic produced by a personal computer, including analysis of the autocorrelation structure. Different distributions were fitted to the interarrival times, packet sizes, and byte count processes with the goal of singling out the ones most suitable for traffic generation.

Categories and Subject Descriptors: C.2.3 [**Computer Systems Organization**]: Computer-Communication Networks: Network Operations—*Network monitoring; Network management*; C.4 [**Computer Systems Organization**]: Performance of Systems—*Measurement techniques*

General Terms: Experimentation, Measurement, Performance

Additional Key Words and Phrases: Modeling, Self-similarity, simulation, source traffic analysis

ACM Reference Format:

Gomes, J. V. P., Inácio, P. R. M., Lakic, B., Freire, M. M., da Silva, H. J. A., and Monteiro, P. P. 2010. Source traffic analysis. ACM Trans. Multimedia Comput. Commun. Appl. 6, 3, Article 21 (August 2010), 23 pages.
DOI = 10.1145/1823746.1823755 http://doi.acm.org/10.1145/1823746.1823755

1. INTRODUCTION

Computer networks underwent a tremendous evolution over the past few years. The network performance and topology have changed and its use spread onto a variety of purposes; even the computer

This research was supported by the Fundação para a Ciência e a Tecnologia, Portugal, through the grant contracts SFRH/BDE/15592/2006, SFRH/BDE/15643/2006 and by the project PTDC/EIA/73072/2006 TRAMANET: Traffic and Trust Management in Peer-to-Peer Networks. It was also funded by Nokia Siemens Networks Portugal S.A., Portugal, by the Institute of Telecommunications, Portugal, and by University of Beira Interior, Portugal.
Authors' addresses: J. V. P. Gomes, P. R. M. Inácio, and P. P. Monteiro, Nokia Siemens Networks Portugal, S. A., Rua Irmãos Siemens, no. 1, 2720-093 Amadora, Portugal; email: {joao.l.gomes, pedro.inacio, paulo.1.monteiro}@nsn.com; B. Lakic, Institute of Telecommunications, University of Aveiro, 3810-193 Aveiro, Portugal; email: branka@av.it.pt; M. M. Freire, IT-Networks and Multimedia Group, Department of Computer Science, University of Beira Interior, Rua Marquês de Ávila e Bolama, 6201-001 Covilhã, Portugal; email: mario@ubi.pt; H. J. A. da Silva, Department of Electrical and Computers Engineering, University of Coimbra, Polo II, 3030-290 Coimbra, Portugal; email: hjas@ci.uc.pt.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 1551-6857/2010/08-ART21 $10.00
DOI 10.1145/1823746.1823755 http://doi.acm.org/10.1145/1823746.1823755

systems acting as network nodes are becoming more and more efficient every day, affecting all underlying communications. Terminal computers working both as server and client through Peer-to-Peer (P2P) applications are shifting the client server paradigm towards a less asymmetric network. The system is almost contents based, demanding more resources and real-time operation. Thus, new studies conducted over recent traffic traces and containing novel traffic classes and applications are required to understand the true behavior of the network of the networks. Although some traffic studies have been made in the past, only a few were based on the traffic generated by an independent source, while most of them focused their efforts to network aggregation points.

During the study reflected herein, we collected and analyzed traces containing traffic from various classes, and tried to fit several known distributions to the experimental distribution, for different metrics of the network traffic. The autocorrelation structure of the process of the bit count per time unit was also subject of analysis because, as stressed in Beran et al. [1995] for video traffic, the correlation embedded in real traffic may be determinant when the need to simulate accurate network conditions arises. Positive correlation influences the way traffic arrives to network devices, impacting resource usage directly. Based on the study results, we present some conclusions that can drive the process of network behavior modeling, simulation and prediction.

The need to generate normal residential and enterprise traffic of a Local Area Network (LAN), for use in a Quality of Service (QoS) assessment project was the first incentive for the work presented herein. Since sometimes the authors faced the situation where the residential LAN was being used by only a single user, responsible for all the network traffic, they had to think of a way to generate traffic at the source level, accurately and in a computationally efficient manner. It is rather difficult to find articles talking about source traffic modeling and simulation based on statistical analysis. Some of them are more interested in how to generate aggregated traffic, defining the sources in respect to that; others define source traffic from a user behavioral point of view that can become complex to use in simulations demanding medium quality source traffic simulation; and most of them do not take into account (nor differentiate) the several types of traffic (namely P2P, Voice over Internet Protocol (VoIP), or others) that come out of a machine, mainly because some of these types are relatively recent. Furthermore, finding all the approaches taken in the past (or at least the most important ones) requires often going through a lot of references and compiling (sometimes diverging) theories.

This article is structured as follows. Section 2 contains a brief overview about traffic modeling and analysis in the past. It summarizes some of the parameters that define source level traffic, when observed in the light of the theories presented in that section. In Section 3, the experimental setup that allowed us to collect the traffic traces is described along with the statistical analysis performed afterwards. The results are then discussed in Section 4, making the analogy to what is said in the literature every time it is possible. The article ends after the main conclusions and future work plans, which are detailed in Section 5.

2. TRAFFIC MODELING AND ANALYSIS IN THE PAST

In this section we try to give a brief overview of what was done till now concerning traffic analysis. We commence by exploring papers regarding the subject of self-similarity and traffic analysis and evolve to the ones describing generation of data, VoIP and Video over IP traffic.

2.1 Network Aggregation Points

First, we introduce some notions that will be used further on in this section. Most of them are taken from Leland et al. [1994].

2.1.1 *Long-Range Dependence, the Hurst Parameter and Heavy Tailed Distributions.* Let $X = \{X(t), t \epsilon \mathbb{N}\}$ be covariance stationary stochastic process with mean μ, variance σ^2 and AutoCorrelation Function (ACF) $r(k)$ ($k \epsilon \mathbb{N}$). Consider also the aggregated processes $X^{(m)} = \{X^{(m)}(i), i \epsilon \mathbb{N}\}$, where $X^{(m)}(i)$ is defined by

$$X^{(m)}(i) = m^{-1}\left(X(i.m) + \cdots + X((i+1).m)\right), \quad \text{for each } m = 2, 3 \ldots. \tag{1}$$

X is said to be *exactly second-order self-similar*, with Hurst parameter $0 < H < 1$, if the law described by (2) holds for *any* $m \epsilon \mathbb{N}$, where $\stackrel{d}{=}$ denotes *equality in distribution*. X is *asymptotically second-order self-similar* if the same condition holds for large m only:

$$X \stackrel{d}{=} m^{1-H} X^{(m)}. \tag{2}$$

It is commonly accepted that *long-range dependence* is the slow power-law decrease of the ACF of a wide sense stationary process (Expression (3)), which happens for $0.5 < H < 1$. The category of X may also be decided recurring to the autocorrelation structure, in which case X is said to be *exactly* or *asymptotically second-order self-similar* depending on wether $r(k)$ and $r^{(m)}(k)$ agree for any $m \epsilon \mathbb{N}$, or for large k and large m only.

$$r(k) \approx k^{2H-2} L(t), \quad \text{where} \quad \lim_{t \to \infty} \frac{L(a.t)}{L(t)} = 1, \forall a \epsilon \mathbb{N}^+. \tag{3}$$

Another definition of great importance in terms of network traffic simulation and analysis is the definition of heavy-tailed distributions. A process $X = \{X(x), x \epsilon \mathbb{R}\}$ is *heavy-tailed* if its distribution function is given by an expression equivalent to

$$P(X > x) = (x/\beta)^{-\alpha}, \quad \text{where} \quad \alpha, \beta \epsilon \mathbb{R}^+. \tag{4}$$

It can be proven [Willinger et al. 1997] that the aggregation of infinite *ON-OFF renewal processes* ruled by a mathematical law from (4) results in a self-similar process with the Hurst parameter given by $H = (3 - \alpha)/2$, where $1 < \alpha < 2$. Some of the theories about the self-similar nature of aggregated traffic draw precisely on the possibility of having heavy-tail distributions governing the way the flows of information are generated/transmitted by the individual (remote) data sources.

2.1.2 *Self-Similarity in Aggregated Network Traffic.* Leland et al. [1994] analyzed several Ethernet LAN traffic traces collected over 4 years, by applying a range of methods for determining self-similarity and the Hurst parameter, so as to build a solid foundation for the claim that the packet/byte count in such network scenario is self-similar. In Willinger et al. [1997], this phenomenon is explained by observing source-destination pairs. The authors mathematically prove that a superposition of many ON/OFF sources exhibiting the *Noah effect* (heavy-tailed distributions of ON and OFF periods) results in traffic exhibiting the *Joseph effect* (self-similarity). Next, they extract source-destination pairs from aggregated traffic to assess whether they can be modeled as such ON/OFF sources. It turns out that they can, with the α parameter of ON periods ranging over the interval $]1, 2[$, and that of OFF periods assuming values from the lower part of that range. The study provides a plausible explanation for the self-similarity of Ethernet LAN traffic.

As for Wide Area Network (WAN)–related traffic, it is said (e.g., in Paxson and Floyd [1995]) that some of its aspects exhibit the properties of asymptotically second order self-similar processes. When trying to explain self-similarity in WAN traffic, scientists [Paxson and Floyd 1995; Willinger et al. 1998] resorted to immigration death process or $M/G/\infty$ queueing model made by Cox [1984]. In this model, M describes the distribution of session arrivals whereas G describes the distribution for the session duration/length. If session arrivals can be modeled by a Poisson process and their duration by

a heavy-tailed distribution, where the increase of packets during a session is a stationary and ergodic process (weaker condition than that of having the packets arriving at a constant rate), the resulting process is asymptotically self-similar. Willinger et al. [1998] claim that the assumption about Poisson process can be replaced by that of the session interarrival times being *independent and identically distributed*. WAN traffic sessions comply with these conditions. The authors study the File Transfer Protocol (FTP), Telnet, and World Wide Web (WWW) related traffic coming to the conclusion that FTP and Telnet arrivals can be described by nonhomogenous Poisson processes, whereas duration of FTP and WWW sessions has a heavy tailed distribution, which ultimately leads to the self-similarity of WAN traffic.

The phenomenon of long-range dependence was first observed in packet byte count. However, the studies of empirical traces that ensued showed that both interarrival times and packet sizes that characterize aggregated traffic appear to be long-range dependent also. Cao et al. [2002] studied the distribution of the packet sizes and of the interarrival times of the aggregated traffic as well and found that the marginal distribution of the interarrival times was matching Weibull distribution best, tending to exponential as the number of connections increases. Minimum average number of connections in a trace was 5.9. This is interesting to mention because it coincides with the results we obtained for various traffic classes when observing traffic generated by a single source. Feldmann et al. [1998] show that, when observed over small time scales, the traffic cannot be described by self-similar models accurately. On such aggregation scales, the influence of protocols and of the network itself becomes visible and determinant, conflicting with the higher level influences (user behavior, distribution of files on the Web, etc.). This network influence exhibits itself through multifractal scaling, but it does not affect self-similarity at bigger time scales.

2.2 Voice over IP, Video, and Data

2.2.1 *Modeling Voice over IP Traffic.* VoIP is a wide spread and still growing service. Models of VoIP traffic are needed to help us to determine the correct measures (packet scheduling, bandwidth reservation etc.) to meet given QoS requirements, such as available bandwidth and time delay.

Duration of calls and call interarrival times are usually described by the exponential distribution, according to a study taken over from the Public Switched Telephone Network (PSTN) [Freeman 2004]. However, findings of an extensive analysis of duration of phone calls performed in Bolotin [1994] do not agree with this approach. The author analyzed over half a million phone calls and determined that the exponential distribution underestimates the frequency of short phone calls, as well as phone calls whose duration is smaller than the mean of the distribution, concluding that the empirically obtained distribution has a longer tail than the exponential distribution. The author shows that a single subscriber holding time for a call can be approximated well with a lognormal distribution and that, for a group of subscribers, a distribution resulting from the combination of three distributions (two lognormal and a uniform distribution) should be used. Packet generation during one call depends on the codec being used and on whether *silence detection* is deployed. In order to save bandwidth, voice activity detectors are used to separate periods during which a person is talking (spurts), from the periods when she or he is silent (gaps). Gaps correspond to listening periods and pauses in speech during which no packets are sent. Initially, both spurts and gaps were modelled using exponential distribution. However, Jiang and Schulzrinne [2000] compare the empirically obtained distributions to the exponential ones and determine that the fit is actually not very good, even when hangover time is used. (Hangover time is used to prevent small gaps; that is, if a gap between two spurts is smaller than the given hangover time, the spurts are merged into one).

In Casilari et al. [2002], the processes describing the duration of spurts and gaps (separated using a threshold distance between two consecutive packets) of a 10-hour videoconference are analysed. The

Table I. Packet Size and Interarrival for Various VoIP Codecs

Codec	Sampling Rate (in Kbit/s)	Time between beginnings of two packets (in msec)	Payload size (in bytes)	Raw packet size (in bytes)
G.711	64	20	160	200
G.729	8	20	20	60
G.723.1	5.3	~45.46	30	70
G.723.1	6.3	~38.46	30	70

Table II. Distributions and Parameters for VoIP Modeling

	Distribution	Mean (in seconds)	
Call duration	Exponential / Lognormal	180–210 (busy hour business environment)	
Call interval	Exponential	4-10	
		Hangover = 20ms	Hangover = 140ms
Spurt	Lognormal	0.326	0.903
Gap	Lognormal	0.442	1.216

authors compare the empirical distributions of the aforementioned processes to various standard distributions (Weibull, Gamma, lognormal, Pareto, and exponential) and come to the conclusion that the best fit for both gap and spurt distribution is the lognormal distribution. In Seger [2003], using Gamma distribution for talk spurts and Weibull distribution for gaps is suggested but not substantiated in any way.

Modeling a phone call requires first deciding on its length (exponential or log-normal distribution). After that, if voice activity detection is used, durations of spurts and gaps must be determined using for example, lognormal distribution. The spurt time is then filled with packets arriving at a constant rate, their size and interarrival time depending on the codec and number of samples per packet. To reduce the inefficiency caused by header overhead, a VoIP packet carries normally between 30 and 160 samples [Seger 2003], depending on time and size constraints. Table I contains the information of 4 different encodings, which may be used to simulate VoIP traffic in a packet-by-packet manner. The values concern CISCO specific encapsulation methods, as discriminated in Cisco Document Server [2002]. As it may be concluded from careful observation of the table, the raw packet size reflects the concatenation of the payload with a 40-bytes Internet header.

Table II summarizes distributions and parameters for modeling VoIP calls. It also presents the information for modeling in-call spurts and gaps, for different hangover values [Jiang and Schulzrinne 2000]. Take into consideration that the call interarrival time process depends heavily on the environment in which the analysis was made [Cisco Document Server 2002]. The exponential distribution is listed in the table because it is the simplest and most widely used model.

2.2.2 *Modeling Video Traffic.* Video coding and transmission can result in Constant Bit Rate (CBR) or Variable Bit Rate (VBR) traffic. CBR coders code all video frames with approximately the same number of bits, which results in some frames being coded at a higher quality than the others while VBR coders code frames with different bit rate achieving the same quality as when using CBR coders but occupying smaller bandwidth [Rose and Frater 1993]. Video traffic consists of three types of data (video, voice, and system data), but most of the models are limited to modeling the video part of the data [Heyman et al. 1992; Reininger et al. 1994; Heyman 1997; Krunz and Makowski 1998; Ansari et al. 2002], namely the size of each video frame in bits or Asynchronous Transfer Mode (ATM) cells, with the video frame being the bit representation of a picture. As the number of frames per second is predefined (25, 30) by the codec itself, one can say that what is indirectly being modeled is the bit rate. It is also possible to find some approaches [Rose 1995a] where a group of pictures is being modeled, instead of one picture at a time.

Table III. Parameters Obtained from the Analysis to the Star Wars Movie Trace.

Type of frame	γ	η	β	μ_0	μ_1
I-frame	4.0605	10.4233	0.4662	0	∼160000
P-frame	1.6605	12.0277	0.3404	0	∼100000
B-frame	1.6431	14.0724	0.3040	0	∼35000

In order to completely capture the effects that video traffic has on a given network, both marginal distribution and ACF of the video trace must be modeled, because models that rely on marginal distribution exclusively underestimate queueing effects such as cell loss and delay. The importance of autocorrelation in video traffic modeling was further backed up by the findings made in Beran et al. [1995], where the authors performed the statistical analysis of a range of video sequences to prove, in the end, that video traffic exhibits long range dependent properties with Hurst parameter varying from 0.6 to 1. The simplest methods for modeling video traces are based on histogram procedures, where the histogram of a real trace is created prior to being used to generate a synthetic trace. Such an approach captures only the distribution of the frame sizes, completely neglecting any requirements related with the correlation between frames.

In order to include the ACF into the model, many different approaches were developed and proposed, which can be roughly divided in Markov chain based models [Maglaris et al. 1988; Sen et al. 1989; Rose 1995a; Hughes et al. 1993], models based on the autoregressive processes [Heyman et al. 1992; Ramamurthy and Sengupta 1992; Dawood and Ghanbari 1999], Transform Expand Sample (TES) models [Reininger et al. 1994; Melamed et al. 1994], M/G/∞ based model [Krunz and Makowski 1998] and models based on self-similar processes [Garrett and Willinger 1994; Liu et al. 1999; Ansari et al. 2002; Huang et al. 1995].

We chose to concentrate on a relatively simple model proposed in Ansari et al. [2002] and Liu et al. [1999]. The authors propose to model an MPEG video trace with three Fractional AutoRegressive Integrated Moving Average (FARIMA) processes (one for each type of MPEG frame) in order to capture self-similarity of a video trace. The mentioned three types of frames are as follows: **I**ntracoded(I)-frames, **P**redictively coded(P)-frames and **B**idirectionally coded(B)-frames [Rose 1995b]. As the Group Of Pictures (GOP) consists usually of 12 frames ordered in the sequence IBBPBBPBBPBB, the only thing one has to assure when generating Video traffic is that the 3 FARIMA processes are alternated to follow that pattern and that the marginal distribution of the frame sizes follows the correct one. According to the same study, the Beta distribution is the one that best describes the said aspect. A simulated self-similar series $\{X_k\}_{k\geq 0}$ (usually following a Gaussian distribution), may then be reshaped into a variable $\{Y_k\}_{k\geq 0}$ following the Beta distribution, recurring to $Y_k = F_\beta^{-1}(F_G(X_k))$, where $F_G(x)$ and $F_\beta^{-1}(x)$ denote the cumulative probability function of the normal distribution and the inverse cumulative probability function of the Beta distribution, respectively (this result may be generalised to any other distribution). The reproduction of a PAL MPEG video requires the generation of 25 frames-per-second.

Ansari et al. [2002] extracted the necessary parameters from a number of different video sequences. For the sake of curiosity, it was decided to include the values for the celebrated and widely used Star Wars sequence in the form of a table (Table III). For other parameters we refer the reader to Ansari et al. [2002]. Notice that the values for μ_1, listed in the last column of the table, were read directly from the corresponding charts in Ansari et al. [2002] and may present small accuracy problems.

The described procedure is suitable for modeling video frame sizes in bits. Transposing such model to the Ethernet frame level requires following the approach from Liu and Babiarz [2007], where 30 frames are generated (instead of 25) and every frame is divided into packets of 1356 bytes of size (for holding 7 188-byte MPEG Transport Stream packets + 40byte IP header). The packets are sent into

Table IV. Distributions and Parameters for ON and OFF Periods for Modeling WWW Related Traffic

Model	Distribution	Parameters
ON times	Pareto	$x_{min} = 1000$ bytes, $\theta = 1.06$
Inactive OFF times	Pareto	$x_{min} = 1$ s, $\theta = 1.5$
Active OFF times	Weibull	$\alpha = 1.46$, $\beta = 0.382$

Table V. Distributions for Modeling Various Aspects of FTP Related Traffic

Model	Distribution
Session interarrival times	Exponential
Connection size	Lognormal
Burst size	Lognormal (distribution body) + Pareto (distribution tail)
Connection interarrival times	0-2.5s exponential; 2.5-4s uniform; 4-180 lognormal

the network at the end of every 33.3 ms interval, excepting for the first time a packet is generated, in which case the frame is sent at a random moment of time in a 33.3ms interval.

2.2.3 *Modeling WWW and FTP Traffic.* Barford and Crovella [1998] developed a tool that mimics the behaviour of a set of users accessing a web server. As a part of their model, they created a user equivalent model of an ON/OFF process. The ON part of the process corresponds to the periods of time a user spends downloading a file, whereas the OFF period of time corresponds to the time between the two downloads. The OFF periods are further divided into active and inactive OFF periods [Crovella and Bestavros 1995], depending on whether they are due to small processing delays inherent to the browsing procedure, or to human interpretation/random assess periods. Table IV summarizes the distributions suggested for three identified subcases. In this table, x_{min} and θ are the location and shape parameters of the Pareto distribution, whereas α and β are the scale and shape parameters of the Weibull distribution.

In Ishac [2001], where FTP related traffic is analyzed, the conclusion that FTP sessions can be described as Poisson process is drawn. During one session there are one or more connections (a connection corresponds to user browsing a directory or downloading a file). Most of the data transferred during one FTP connection is clumped into bursts (connections following one after another shortly), which is a phenomenon that finds an explanation in the users themselves, as they commonly choose more than one file to download at once. Transfer rate during one connection varies due to the Transmission Control Protocol (TCP) congestion avoidance algorithm - before sending a new packet, server must wait for acknowledgement of reception. The distributions describing different aspects of FTP traffic are summarized in Table V.

3. SOURCE TRAFFIC ANALYSIS AND MODELING

The initial phase of our study consisted of collecting data samples. For that purpose, traffic from several different classes was captured in several network environments, representing various contexts and scenarios of typical usage of the Internet by a home or enterprise user. With the objective of building a model for each traffic class, we considered three processes that describe a trace of data: the byte count per time unit, the interarrival times and the packet sizes. These processes were then statistically analyzed. Section 3.1 explains how the data was collected, providing a detailed description of each considered scenario. Each traffic trace is described in Section 3.2, while Section 3.3 focuses on their analysis and on the fitting procedure.

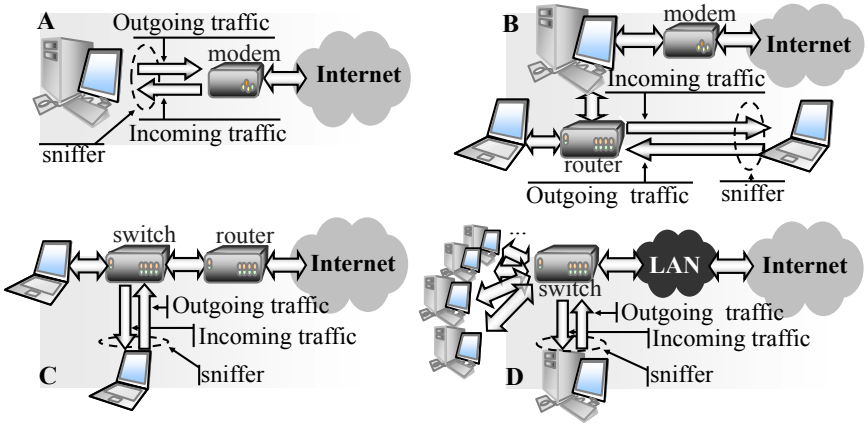

Fig. 1. Logical placement of the traffic sniffer, during the data collecting procedure for the first (a), second (b), third (c), and (d) fourth scenarios.

3.1 Experimental Setup

Since the main goal of this study was to analyze source traffic, the traces were collected directly on an end-user machine. We used a sniffer to capture the traffic into a TCPDUMP [2008] format file. Additionally, the traces were divided into incoming and outgoing traffic, generating two more files, producing a total number of three trace files for each scenario. For the sake of simplicity, IN and OUT will be used to refer to the INcoming and OUTgoing subset of data, respectively, and the main data file will be referred to as MIX. All the traces were collected in one of four different network scenarios, described in the following four sections.

3.1.1 *First Scenario: A Single Machine Connected Directly to the Internet.* The first scenario consisted of a desktop computer, running Windows XP Operating System (OS), with an Intel Pentium 4 processor running at 1.7GHz and 512MB of Random Access Memory (RAM). The computer uses an Ethernet board performing at 100Mbit/s, connected to a commercial 2Mbit/s Internet connection. This first scenario and the logical placement of the packet sniffer are depicted in Figure 1(a).

3.1.2 *Second Scenario: Small Network Connected to the Internet through a Gateway.* In the second scenario, a small network formed by two laptops connected to a router was built. The first computer was running Windows XP OS and the second one was running a Linux OS distribution. The router was connected to a commercial 2Mbit/s Internet link through a desktop computer running Windows XP and working as an Internet gateway. Figure 1(b) complements the explanation given for this scenario.

The main difference between the present and the previous scenarios lies in the way the computers are connected to the Internet. While, in the first scenario, the computer was directly connected to the Internet, using a public IP address in its communications, in the second one, it suffers the effects of Network Address Translation (NAT) mechanisms and shares the Internet access with another active computer. The traces were captured in the laptop running Windows XP, with a Intel Core 2 Duo 1.80GHz processor and 1024MB of RAM, using an Ethernet board performing at 100Mbit/s to connect to the network.

3.1.3 *Third Scenario: Small Network Connected to the Internet through a Router.* In the third scenario, two laptops were connected to a switch, which was connected to a commercial 2Mbit/s Internet connection via a router, as depicted in Figure 1(c). Both computers were running Windows XP OS. Such

combination is rather common in countries where the Internet connection is being distributed using Digital Subscriber Lines (DSL) or cable, where a single access connection is often shared between residents of an apartment or set of apartments, using routers and switches. The traces were captured on a laptop with an Intel Centrino 1.73GHz processor and 1024MB of RAM, connected to the network through an Ethernet board performing at 100Mbit/s.

3.1.4 *Fourth Scenario: Final Branch of a Big Local Network.* In the fourth scenario, the final branch of a large local area network was observed. The traces collected are those exchanged between a computer and the closest switch of a complex tree structure that, ultimately, connects to the Internet via a 1Gbit/s Internet connection. The computer used was a desktop system with an Intel Pentium 4 2.80GHz processor and 512MB of RAM. It was connected to the local network using an Ethernet board at 1Gbit/s. Figure 1(d) schematizes the described scenario.

3.2 Description of the Traces

To enable a granular analysis of all the traffic types that we could think of at the time the traces were made, we used different telematic applications during the collecting period and categorised the traces according to the software that generated them, and to the scenario where they were collected, as depicted by Table VI. Characteristics such as the size of the trace and the number of captured packets can also be found in the same table.

The emphasis was put on relatively new types of traffic (e.g., P2P, streaming, VoIP). The reasons behind this were given in the introduction section of this document and are mostly related to the popularity the respective applications gathered among residential (and even corporate) users. It should be mentioned that all the broadcast traffic was filtered out from the trace captured in the last scenario. This was done to avoid having the analysis biased by unwanted messages generated by (too many) other computers in the network.

3.3 Fitting Distributions and Studying Autocorrelation

Processes describing packet sizes, interarrival times and byte count of each of the data files listed in Table VI were constructed and automatically processed using a custom-made JAVA script that puts together the cumulative probability function for each of them and adjusts the parameters of several theoretical distributions, as described in the following section. The preselection of the model that best fits the empirical results is made by comparing the discrepancy D^k_{max} between each of the theoretical cumulative distributions $F^k_t(x)$ and the empirical one $F_e(x)$, where $D^k_{max} = \max_{x_i \epsilon \Omega} |F^k_t(x_i) - F_e(x_i)|$ and Ω is the set of incidences of the studied sequence of values (consider observing the three first charts of Figure 4 for a graphical representation of this metric). The one presenting the smallest value for that particular metric is momentarily labeled the most suitable one. The decision is then supported by human discernment, so as to filter out cases where even the best among all the choices does not fit the data in a satisfactory manner. Notice that the metric defined (D^k_{max}) is the same used in the Kolmogorov-Smirnov goodness of fit test.

Since the autocorrelation structure of the bit count per time unit may be of critical importance for the analysis of queueing effects in network aggregation points, it was decided to complement the analysis of the three previous traffic aspects with additional comments on that statistical property. The main goal of this analysis is to provide potential practitioners with the idea of the amount of dependence that needs to be simulated when modeling source traffic, rather than to provide the exact expression of the ACF for each type of traffic. Constrained by the length of some of the traces and for the sake of coherence, the calculation of the autocorrelation is made only for lags k smaller than 40s ($k = 1, 2, 4, \ldots, 40$). The ACF is denoted herein by $r(k)$, and its assessment is made recurring to

Table VI. Description and Characterization of the Collected Traces

Traffic trace		Duration (in minutes)	Scenario	Size (in bytes)	Number of packets	Date
Web	OUT	15	3	366999	1964	30-10-2006
	IN			2335860	2483	
	MIX			2702859	4447	
Skype VoIP	OUT	11	1	1820367	12582	12-11-2006
	IN			1818953	12637	
	MIX			3674422	25309	
Streaming download	OUT	27	3	3297385	1964	30-10-2006
	IN			170588472	115525	
	MIX			174002976	177966	
Streaming broadcast	OUT	41	3	269343	3251	30-10-2006
	IN			12767549	23190	
	MIX			13211419	29015	
eMule (file upload)	OUT	3	1	1316487	2120	12-11-2006
	IN			120456	1743	
	MIX			1455207	3910	
eMule (file download)	OUT	13	1	1473419	17873	12-11-2006
	IN			11198338	18722	
	MIX			12734669	36757	
File download from web	OUT	11	3	1077621	19334	30-10-2006
	IN			35993801	24065	
	MIX			37084869	42590	
MSN VoIP	OUT	8	1	2968328	38144	12-11-2006
	IN			1464972	13320	
	MIX			4481692	37188	
Mail, MSN and file sharing traffic	OUT	41	2	10120097	35388	19-02-2007
	IN			54569377	48835	
	MIX			64689834	84229	
File Sharing, download from web and MSN traffic	OUT	54	2	23140782	113454	19-02-2007
	IN			182210157	156944	
	MIX			205351179	270402	
File Sharing, streaming download and MSN traffic	OUT	39	2	18517605	52263	19-02-2007
	IN			45412410	56533	
	MIX			63931435	108812	
Skype, streaming download and MSN traffic	OUT	67	2	36133420	128737	19-02-2007
	IN			22580277	94461	
	MIX			58714647	223210	
Web, mail and instant messaging traffic	OUT	1261(∼21h)	4	78862681	253076	14-06-2007
	IN			263395433	313684	
	MIX			388818159	1039717	

$r(k) = \overline{X_t \times X_{t+k}}$, where $\overline{X}$ denotes the *average* of the series represented by X. Prior to the calculation, all the empirical sequences concerning the bit count per time unit were dully normalised and further processed to reduce possible trends and lacks of stationarity, and allow direct application of the aforementioned formula. The results of this particular analysis were summarized in the form of a 2-tuple in the charts of Figures 2, 3 and 4, and in the last column of Table VII. The pair consists of the maximum and of the minimum values of $r(k)$ for $k = 1, 2, 4, \ldots, 40$.

In the following sections, we describe the study performed on each type of traffic in more detail and discuss some of the results. During the analysis, a set of at least 3 charts was made for every available trace. However, to avoid extending the paper excessively, we refrained from presenting all of them. To

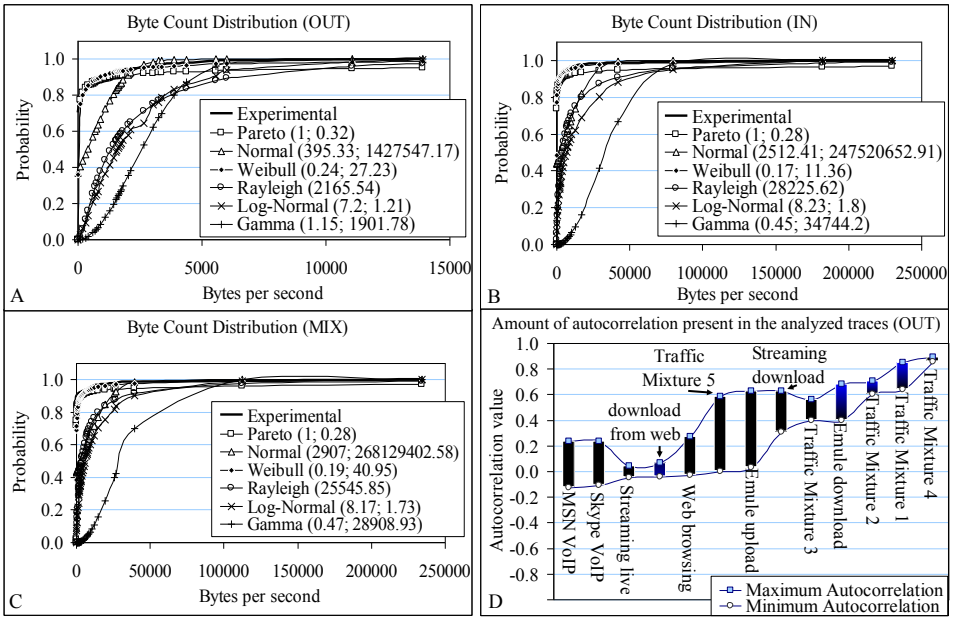

Fig. 2. Cumulative probability functions for the byte count per time unit process, for (a) outgoing traffic, (b) incoming traffic, and (c) incoming plus outgoing traffic. The empirical data under analysis concerns the traces of Web Traffic Without Streaming. (d) Variation interval of the first 40 values of the ACFs of the byte count per time unit process, calculated for the OUT datasets.

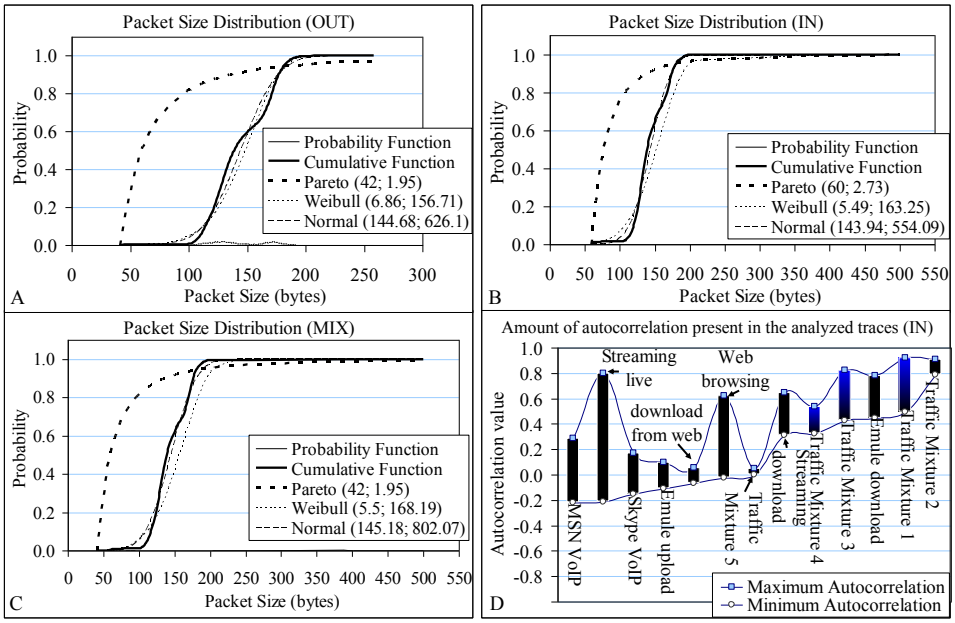

Fig. 3. Probability and cumulative functions of the packet size distributions for (a) outgoing traffic, (b) incoming traffic and (c) incoming and outgoing traffic. The empirical data under analysis concerns the traces of Skype VoIP Traffic. (d) Variation interval of the first 40 values of the ACFs of the byte count per time unit process, calculated for the OUT data sets.

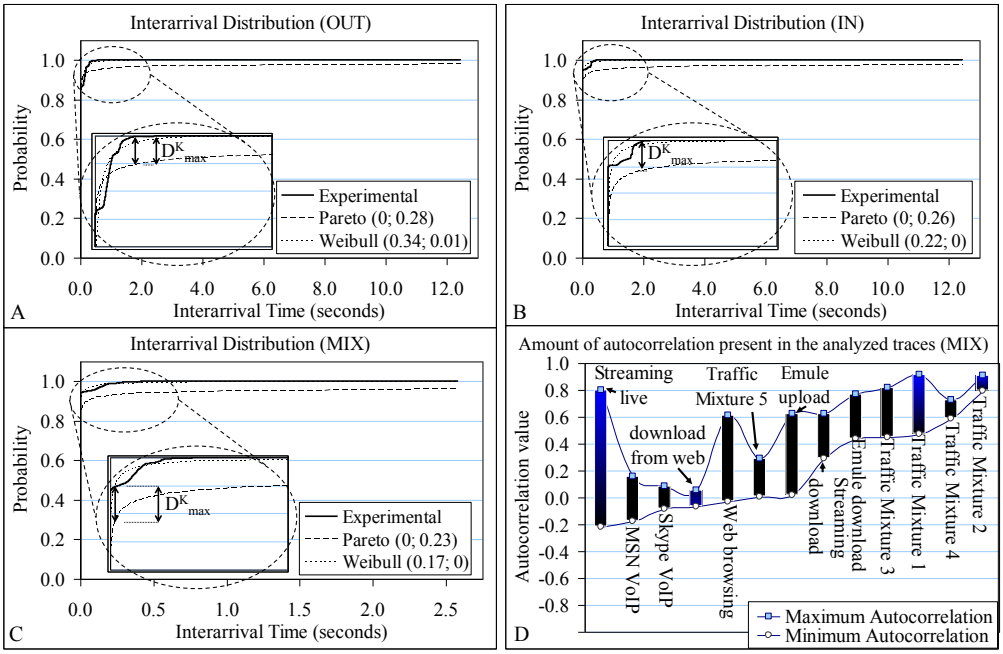

Fig. 4. Cumulative probability functions of the interarrival time process for (a) outgoing traffic, (b) incoming traffic and (c) incoming plus outgoing traffic of streaming download relative traffic. The empirical data under analysis concerns the traces of Streaming Broadcast Traffic. (d) Variation interval of the first 40 values of the ACFs of the byte count per time unit process, calculated for the MIX data sets.

explain and exemplify the statistical analysis and the fitting process, we chose to exhibit only one set of charts for each of the first three traffic classes and keep the remaining descriptions textual. We then summarise all the results, in a more informal way, in Section 4. Prior to the analysis itself, we are going to briefly introduce mathematical models used in this work and explain methods used to estimate the parameters for each distribution.

3.3.1 *Models and Parameter Estimation.* In order to find the most suitable theoretical models for the previously mentioned processes, we tried to shape several well-known distributions by estimating their specific parameters. Inspired by the literature and by some of the preliminary observations, we opted for the following ones.

—The *Pareto distribution*, which is defined by the typically termed *location* (x_{min}) and *shape* (θ) parameters, and that will be referred to as $P(x_{min}, \theta)$ during the remaining part of this document. Its probability function is given by (5), and its parameters can be assessed using the formulas in (6), where Ω denotes the set with N incidences of the empirical process:

$$P(X = x) = \theta x_{min}^{\theta} / x^{\theta+1}, \quad \text{with} \quad x \epsilon [x_{min}; +\infty[\text{ and } x_{min}, k \epsilon \mathbb{R}^{+}; \tag{5}$$

$$x_{min} = \min_{x_i \epsilon \Omega}(x_i) \text{ and } \theta = N \left(\sum_{i=0}^{N} (\ln x_i - \ln x_{min}) \right)^{-1}, \; x_i \epsilon \Omega. \tag{6}$$

—The *Normal distribution*, commonly defined by the well known probability function (7), which will be denoted herein as $N(\mu, \sigma^2)$, with the μ and the σ^2 denoting the expected value and the variance

Table VII. Summary of the Results Obtained for the Several Traffic Aspects

Traffic Trace Data Set		Byte Count (in bytes per second)	Packet Size (in bytes - max. value 1516)	Interarrivals (in seconds)	ACF variation interval
Web without Streaming	OUT	W(2.42E-1, 27.23)	W(3.73E-1, 50.95)	W(3.80E-1,5.25E-2)	[-0.032, 0.271]
	IN	W(1.68E-1, 11.36)	60(26%), ≥1484(52%)	W(3.81E-1, 2.65E-2)	[-0.024, 0.623]
	MIX	W(1.93E-1, 40.95)	54-62(48%), ≥1484(30%)	W(2.50E-1, 5.33E-3)	[-0.028, 0.614]
Skype VoIP	OUT	W(10.16, 5099.29)	N(144.68, 626.1)	W(5.88E-1, 5.89E-2)	[-0.109, 0.234]
	IN	W(11.82, 5063.86)	N(143.94, 554.09)	W(2.99, 3.40E-2)	[-0.151, 0.174]
	MIX	W(21.38, 10047.74)	N(145.18, 802.07)	W(1.29, 1.66E-2)	[-0.086, 0.086]
Streaming Download	OUT	N(2025.45, 399485.72)	54(99%)	W(3.44E-1, 6.23E-3)	[0.306, 0.626]
	IN	N(104784.07, 1006.94E+6)	1514(95%)	W(2.20E-1, 3.62E-4)	[0.313, 0.643]
	MIX	N(106554.18, 1115.16E+6)	54(34%), 1514(62%)	W(1.73E-1, 4.90E-5)	[0.294, 0.627]
Streaming Broadcast	OUT	W(1.04, 108.39)	45-84(93%)	See Section 3.3.4	[-0.048, 0.042]
	IN	N(5084.77, 19485069.77)	517-526(88%)	P(8.00E-6, 2.75E-1)	[-0.215, 0.801]
	MIX	N(5257.23, 19611722.80)	45-84(23%), 517-526(71%)	W(6.62E-1, 5.76E-2)	[-0.217, 0.800]
eMule (File Upload)	OUT	W(3.04, 6422.23)	W(4.82E-1, 703.99)	W(4.63E-1, 9.34E-2)	[0.029, 0.624]
	IN	W(2.48, 577.73)	W(1.27, 47.45)	W(8.49E-1, 1.33E-1)	[-0.109, 0.095]
	MIX	W(3.32, 7002.00)	W(4.15E-1, 208.44)	W(4.27E-1, 3.20E-2)	[0.017, 0.627]
eMule (File Download)	OUT	W(1.18, 1968.07)	W(6.77E-1, 43.11)	W(6.82E-1, 3.57E-2)	[0.399, 0.682]
	IN	N(14033.01, 97066134.82)	60-72 (40%), ≥1494 (23%)	W(8.90E-1, 3.74E-2)	[0.447, 0.778]
	MIX	N(15958.23, 110.16E+6)	W(2.60E-1, 92.68)	W(5.09E-1, 1.18E-2)	[0.438, 0.768]
File Download from Web	OUT	N(1550.54, 160774.45)	54-74(99%)	W(3.51E-1, 1.53E-2)	[-0.045, 0.067]
	IN	N(51789.56, 190653714.80)	1514(99%)	W(3.27E-1, 8.53E-3)	[-0.068, 0.057]
	MIX	N(53370.46, 199739532.26)	54-74(44%), 1514(56%)	W(1.86E-1, 4.12E-4)	[-0.066, 0.057]
MSN VoIP	OUT	N(5456.51, 2940917.82)	W(8.77, 128.26)	W(1.67, 2.75E-2)	[-0.129, 0.236]
	IN	N(2693.21, 6356671.55)	W(4.99, 133.54)	W(1.19, 2.81E-2)	[-0.219, 0.287]
	MIX	N(8238.40, 4854388.43)	W(5.04, 141.32)	W(1.22, 1.74E-2)	[-0.177, 0.160]
Mail, MSN, and File Sharing Traffic	OUT	W(3.76E-1, 2546.25)	54-77(77%), ≥1402(14%)	W(4.68E-1, 2.53E-2)	[0.641, 0.851]
	IN	W(3.11E-1, 5230.75)	60-62(10%), 1099(15%), ≥1484(61%)	W(5.02E-1, 1.96E-2)	[0.493, 0.922]
	MIX	W(3.99E-1, 10338.75)	54-77(40%), 1099(9%), ≥1402(41%)	W(3.23E-1, 4.07E-3)	[0.471, 0.917]
File Sharing, Download from Web, and MSN Traffic	OUT	N(7050.82, 30199712.90)	54-74(88%), ≥1414(10%)	W(6.31E-1, 2.17E-2)	[0.605, 0.702]
	IN	W(8.69E-1, 44999.36)	60-66(6%), 1514(69%)	W(7.08E-1, 1.82E-2)	[0.787, 0.907]
	MIX	W(9.34E-1, 57074.99)	54-74(41%), ≥1414(45%)	W(5.13E-1, 6.47E-3)	[0.792, 0.906]
File Sharing, Streaming Download, and MSN Traffic	OUT	N(7728.56, 30210101.80)	54-77(73%), ≥1506(17%)	W(5.68E-1, 3.02E-2)	[0.396, 0.560]
	IN	W(1.19, 19625.19)	60-66(19%), ≥1506(43%)	W(7.29E-1, 3.26E-2)	[0.426, 0.824]
	MIX	W(1.26, 29481.66)	54-77(47%), ≥1506(30%)	W(4.96E-1, 1.06E-2)	[0.451, 0.818]
Skype, Streaming Download, and File Sharing Traffic	OUT	N(8873.73, 36352160.02)	P(42, 8.93E-1)	W(7.70E-1, 2.69E-2)	[0.855, 0.889]
	IN	N(5546.58, 40574100.44)	P(60, 1.13)	W(9.99E-1, 4.05E-2)	[0.322, 0.539]
	MIX	N(14419.21, 88421057.85)	P(42, 8.82E-1)	W(6.81E-1, 1.21E-2)	[0.585, 0.729]
Web, Mail, and Instant Messaging Traffic	OUT	W(2.65E-1, 30.01)	54-62(61%), ≥1482(12%)	W(2.40E-1, 2.02E-2)	[0.003, 0.330]
	IN	W(1.73E-1, 2.28)	60-93(28%), ≥1506(49%)	W(2.19E-1, 1.17E-2)	[-0.002, 0.024]
	MIX	W(3.27E-1, 285.35)	54-93(62%), ≥1482(18%)	W(4.00E-1, 3.96E-2)	[0.003, 0.149]

of the analysed process, respectively. The parameters of the Normal distribution can be estimated using the formulas in (8), with p_i being the relative frequency of event x_i:

$$P(X=x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad \text{with } x\epsilon\mathbb{R}, \mu\epsilon\mathbb{R} \text{ and } \sigma > 0; \tag{7}$$

$$\mu = \sum_{i=0}^{N} p_i \times x_i \quad \text{and} \quad \sigma^2 = \sum_{i=0}^{N} p_i \times x_i^2 - \mu^2, \quad \text{with } N = |\Omega| \quad \text{and} \quad x_i\epsilon\Omega. \tag{8}$$

—The *Lognormal distribution*, defined by a simple transformation of (7) resulting in Expression (9) and by the same parameters as Normal distribution. The values of μ and of σ^2 may be obtained using the formulas in (10), where $\bar{X}$ and $Var(X)$ denote the average and the variance of the

experimental trace:

$$P(X=x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}, \quad \text{with} \quad x\epsilon\mathbb{R}^+, \mu\epsilon\mathbb{R} \quad \text{and} \quad \sigma > 0; \tag{9}$$

$$\sigma^2 = \ln\left(1 + \frac{Var(X)}{\overline{X}^2}\right) \quad \text{and} \quad \mu = \ln(\overline{X}) - \frac{1}{2}\sigma^2. \tag{10}$$

—The *Weibull distribution*, represented by the probability function (11) and denoted herein by $W(\beta, \alpha)$. The maximum likelihood estimator of α (the scale) and β (the shape) relies on solving the equation in (12) for β (e.g., using an iterative method like the Newton-Raphson method), and by then setting $\alpha = N^{-1}\sum_{i=1}^{N}\ln(x_i^\beta)$:

$$P(X=x) = \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} e^{(-x/\alpha)^\beta}, \quad \text{with } x\epsilon\mathbb{R}^+ \text{ and } \alpha, \beta > 0; \tag{11}$$

$$\frac{\sum_{i=1}^{N} x_i^\beta \ln(x_i)}{\sum_{i=1}^{N} x_i^\beta} - \frac{1}{\beta} - \frac{1}{N}\sum_{i=1}^{N}\ln(x_i) = 0 \quad \text{, where} \quad N = |\Omega| \text{ and } x_i\epsilon\Omega. \tag{12}$$

—The *Rayleigh distribution*, which is described by its probability function (13) and its only parameter ω. The maximum likelihood estimator for ω is given by (14):

$$P(X=x) = x\frac{e^{-\frac{x^2}{2\omega^2}}}{\omega^2}, \quad \text{with } x\epsilon\mathbb{R}^+ \text{ and } \omega > 0; \tag{13}$$

$$\omega = \sqrt{\frac{1}{2N}\sum_{i=1}^{N} x_i^2}, \quad \text{where } N = |\Omega|. \tag{14}$$

—The *Gamma distribution*, whose probability function follows (15) (where $\Gamma(k)$ is the Gamma function), is uniquely characterised by two parameters, k and δ. Both parameters have well documented maximum likelihood estimators that can be summarized by Expressions (16) and (17):

$$P(X=x) = x^{k-1}\frac{e^{-x/\delta}}{\Gamma(k)\delta^k}, \quad \text{with } x > 0 \text{ and } \delta, k > 0; \tag{15}$$

$$k = \frac{3 - s + \sqrt{(s-3)^2 + 24s}}{12s}, \quad \text{with } s = \ln\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) - \frac{1}{N}\sum_{i=1}^{N}\ln(x_i), \tag{16}$$

$$\delta = \frac{1}{k \times N}\sum_{i=1}^{N} x_i, \quad \text{where } N = |\Omega|. \tag{17}$$

3.3.2 *Web Traffic without Streaming.* The traffic class one cannot forget when talking about source traffic analysis is Hypertext Transfer Protocol (HTTP)-related traffic, herein baptized as Web Browsing. This class contains all the packets generated by browsing activities from a given network user. Although this type of communications should also include file downloads, webcast and streaming (e.g., radio streaming)-related traffic, we decided to analyze that types of traffic separately (in a subsequent

section), as they result in slightly different transmission profiles that deserve to be addressed independently.

The set of charts chosen to represent Web-related traffic concerns the analysis conducted for the byte count per time unit of this particular collection of traces. Figures 2(a), 2(b), and 2(c) contain the plots of the byte count per time unit cumulative distribution function. These figures show that Weibull is the model that best fits the bit rate of Web-related traffic. From the three datasets, the cumulative distribution of the outgoing portion of the traffic is the one presenting the most irregular shape (and bigger discrepancy values), which is mainly motivated by the well-known asymmetry of this type of traffic. In the referred direction, communications are dominated by silence periods interleaved with sporadic HTTP requests, SYN or ACKs packets, resulting in relatively small and less varying transmission rates (average of 394.62($\pm$76.85) bytes-per-second, with an estimated standard deviation of 1195.68($\pm$581.96)).

The chart in Figure 2(d) plots the minimum and the maximum values of the first 40 ACF incidences (i.e., $min_{k=1,2,...,40}(r(k))$ and $max_{k=1,2,...,40}(r(k))$, respectively), against the designation of the traffic under analysis. Notice that the aforementioned chart concerns the study of the OUT data sets only, and that similar graphical representations for the IN and MIX traces are included in the following two subsections, to save space. Note also that the order by which the several classes appear in the chart is decided upon the minimum values, which are sorted in an increasing manner, and that the designations *Traffic Mixture 1, 2, 3, 4 and 5* are used to refer the traces containing more than one traffic type, being those discussed afterwards. The ACF is defined for the interval $[-1.0, 1.0]$, and it is typically said that, for values bigger than 0, the process is positively correlated, while it the contrary case, it is negatively correlated or anticorrelated.

The variation intervals of the ACF are never positioned in the lower part of the charts, in any of the communication directions. Thus, none of the traces exhibits significant anti-correlation. The label Web browsing appears in the 5th position of Figure 2(d) and Figure 4(d), and in the 7th position of Figure 3(d). According to the results, the outgoing communications are weakly correlated, as a reflection of the unpredictable manner by which a human interacts with the WWW, while the IN dataset presents higher autocorrelation values, influenced by the network functioning and length of the connections.

As for the packet size process, we came to the conclusion that it would be possible to roughly model the outgoing traffic of a source using Weibull also, shaped with the parameters 3.75E-1($\pm$7.96E-3) and 50.95($\pm$2.81). Unfortunately, the same does not apply to the other two trace files (IN and MIX). Actually, none of the distributions used was able to fit the empirical distributions of the packet size process. The reason behind this fact lies in their specificity in terms of probability peaks, which will be mentioned in almost all subcases that follow.

The results prove that Weibull constitutes the best choice for modeling the interarrival times process as well (please observe the table in Section 4.1 for details on the parameters of the distribution). It is important to mention that similar analysis of HTTP traffic obtained from a WAN link was already made in Casilari et al. [2004]. They tried to fit several distributions to the empirical packet size and interarrival distributions, but singled Pareto distribution out as the best fit for the interarrival times.

3.3.3 *Skype VoIP Traffic.* The second set of traces analyzed concerns VoIP related traffic, namely Skype traffic. Skype generates variable bit rate data, which is primarily reflected on the packet size process and ultimately, in the byte count per time unit. The first conclusion we are going to point out is that the process describing the byte count per time unit is well modeled by Weibull, but that it can also be (fairly) modeled by the Gamma, Normal, and Lognormal distributions. As for the packet size distribution, please observe Figures 3(a), 3(b), and 3(c), comprising the set of charts chosen to represent this particular study case. The theoretical model that best describes the aforementioned

process is the Normal distribution, even though Weibull distribution provides a good fit as well. The Normal distribution is centered around the size of 144 bytes for the three analyzed cases, presenting a standard deviation of approximately 28 bytes for the incoming traffic and of approximately 23 bytes for the outgoing traffic. This Gaussian variation results from the combination of several factors (the way humans dialogue, the codification of the voice, and the means by which Skype sends packets into the network) and is actually a curious fact to notice, as the packet sizes are often difficult to model recurring to theoretical distributions. In most of the studied cases, the best way of modeling such aspect would be to use a previously recorded (empirical) distribution.

In terms of interarrival times, there is not much to say as the conclusions drawn before for the reciprocal process of Web-related traffic applies to this study case also. Weibull fits the empirical distribution for all the three cases. On the other hand, the same cannot be said in terms of autocorrelation structure of the traces. According to the investigation, only the outgoing portion of the traffic is lightly correlated for a few number of lags, presenting the variation interval $[-0.109, 0.234]$. The autocorrelation analysis for the files containing the Skype IN and MIX communications produced balanced variation intervals of $[-0.151, 0.174]$ and of $[-0.086, 0.086]$, respectively, uncovering the unpredictable nature of this kind of traffic (when silence detection is used, VoIP depends significantly of human behavior). The set of results concerning the analysis of the autocorrelation for the IN data sets is concatenated in Figure 3(d), following an analogous procedure to the one mentioned in the previous section for the same subject. Notice that the predominance of positive correlations is once more flagrant in this chart, and that the Skype label is the 3rd from the left, associated with one of the smallest variation intervals in the chart.

3.3.4 *Streaming Traffic.* As previously mentioned, two different embodiments of streaming related traffic were studied in the context of this work: the streaming download of a file and the reception of a live streaming broadcast. For the former case study, the results lead to the conclusion that the processes describing the byte count per time unit and the interarrival times could be either modeled by the Normal or by the Weibull distribution, respectively, with parameters 2025.45($\pm$30.71) and 399485.72($\pm$51292.78) for the first, and 3.44E-1($\pm$4.71E-3) and 6.23E-3($\pm$3.15E-4) for the second (values concern the outgoing traffic only). As for the packet size series, the results confirm our expectations. None of the considered models suits such process, because its discrete probability function is dominated by the peaks around 1500 bytes, for the incoming trace, and around 50 bytes, for the outgoing trace. These values correspond to the maximum IP packet size over Ethernet and to the size of common TCP acknowledgements.

Figures 4(a), 4(b), and 4(c) depict the fitting process conducted for the interarrival times of streaming download traffic and Figure 4(d) contains the plot that summarises the study of the autocorrelation structure of the trace, constructed in the same way as previously indicated (but for the MIX data sets). According to the plot, all communications exhibit positive autocorrelation till the aggregation scale of 40s, with values close to 0.425 for the three data sets (streaming download is the 8th label in all the ACF charts), inspired probably by the constancy introduced by the streaming protocol and by the packet size morphology.

In the (not included) charts concerning live streaming broadcast, we observed a probability peak around 8000 bytes-per-second, for the incoming communications. The reason behind this fact is, obviously, the streaming protocol itself, which secures its reliability with the constant bit rate specification. Because of that, the byte count per time unit process for both the incoming communications is difficult to model. As the incoming packets tend to dominate the overall communications for this type of application, the trace containing the complete trace suffers from the same problem. Most packets of the outgoing communications have between 50 and 100 bytes (ACKs), while all incoming ones fall within

the range 517 and 526 bytes, irregularly concentrating the probabilities in the said intervals. Another curiosity we noticed about this type of traffic was that the time interval of 1 second is rather common in the outgoing traffic. This is again due to a regular protocol feedback mechanism (e.g., acknowledgement), which ultimately impacts the distribution of the packet sizes in such a way that none of the considered models can satisfactorily fit the interarrival times in the said traffic direction. The other two subsets can be reproduced using Pareto and Weibull, Pareto being the most suitable for the downstream direction and Weibull the best for the traffic MIX. This type of streaming traffic may thus be reproduced by randomly selecting 88% of the packet sizes between 517 and 526 bytes and by creating interarrivals according to $P(8E - 6, 2.75E - 1)$, for the incoming traffic, and by simulating a packet smaller than 84 bytes every 1 second, for the outgoing traffic. The ACF of streaming broadcast decreases rapidly, as the time lag increases, but presents the biggest variation interval of all IN and MIX data sets. According to the analysis, the outgoing connections are random.

3.3.5 *eMule File Sharing Traffic.* The second P2P related traffic examined in the scope of this work was eMule traffic, generated during common file sharing. We collected traces for the scenario where several files are being downloaded simultaneously and for the one where at least one file is being sent to a group of P2P users. Both situations were disjoint, that is, no files were being upload during the collecting stage of file downloads, and vice versa.

According to the analysis, the communications related to file upload using eMule can be modeled by the Weibull distribution, even though it performs badly for incoming traffic. It has to be said that the data scarcity (compared to the outgoing data flow), in the said direction, affects the statistical analysis severely. The observation of the (not included) ACF lets us know that the dependence embedded in the bit rate per time unit of outgoing traffic is of short range (i.e., autocorrelation values tend to 0, as the time lag increases).

On the other hand, downloads from multiple sources result inevitably in highly dynamic incoming transmissions with a variable bit rate. The autocorrelation values are definitely higher than their counterparts for all datasets, and Weibull is the best choice for modeling the interarrival times and the packet size distributions, except for incoming traffic (in which case, it was again not possible to model the packet sizes). The distributions that better describe the byte count per time unit are the Weibull for the outgoing communications, and the Normal for the other two (though Rayleigh embodies a fine choice also). Again, notice that the values estimated for the aforementioned distributions can be found in Table VII.

3.3.6 *File Download from Web.* The process assigned to packet sizes of the traffic created by Web download could not be satisfactorily modeled by any of the considered distributions, while the interarrival and byte count processes follow the mathematical laws of Weibull and Normal distributions, respectively. Almost all the packets in downstream have 1500 bytes, whereas those in upstream are 50 bytes long (TCP acknowledgements). That is understandable, since the biggest part of a downloaded file is transferred using the biggest available data units, corresponding to 1500 byte long IP over Ethernet frames.

The byte count process of the type of traffic under observation behaves randomly, as proven by the autocorrelation values, which do not deviate from 0 more than 0.07 units. Since the packet sizes are almost constant during a download, the explanation for this fact lies in the interarrival times, impacted by the effects in the network nodes between the server and the client. Network influence renders the byte count per time unit unstable and mostly unpredictable, contrary to what was happening with the streaming traffic, which has to comply with stricter QoS requirements.

3.3.7 *Messenger VoIP Traffic.* MSN VoIP traffic constitutes a second example of VoIP traffic included in our analysis. Surprisingly, the aspect that was most difficult to model was the byte count process, which could only be approximated poorly using Weibull, Normal, Lognormal, and Gamma distributions. Among them, we emphasize the Normal distribution for being the one that models the incoming traffic trace best. The autocorrelation values show that none of the data sets exhibit significant positive or negative dependencies (please observe Table VII). The processes concerning the packet size and the interarrival times were easily modelled using Weibull for all the considered data sets.

3.3.8 *Traffic Mixture 1: Mail, Messenger, and File Sharing.* After investigating some traffic classes separately, it was decided to move to the scenarios that can reflect the behavior of a contemporary Internet user, by constructing a trace containing a mixture of traffic classes, generated by Mail, messenger, and file sharing applications. According to our findings, the Weibull distribution is once more the one that approximates the probabilistic laws of the byte count per-time-unit process best. The autocorrelation of the referred process is the highest of all the analyzed traces, with values ranging in the interval $[0.641, 0.851]$ for the outgoing and $[0.493, 0.922]$ for the incoming communications. The packet size continued to respond badly to fitting procedure and Weibull proved itself once again the best option for modeling interarrival times.

3.3.9 *Traffic Mixture 2: File Sharing, Web Download and Instant Messaging.* This section describes the case where a user downloading several files via a P2P network is using the instant messenger to (text) chat with someone. Additionally, this person is using the browser to download a file from a Web server. While, in this case, the Normal distribution can be safely used to model the byte count per time unit of the OUT dataset, none of the considered distributions were suitable for modeling the other aspects of traffic. If forced to choose one of them, it would have to be Weibull, since it is the one that gives the best approximation of the empirical data. The autocorrelation embedded in the byte count per time unit is again extremely high and positive (values ranging in the interval $[0.605, 0.907]$), primarily due to the presence of file sharing traffic (as mentioned earlier) and secondly due to the interaction of the Web download with the remaining types of traffic. Alone, the, Web download does not exhibit significant autocorrelation, but when mixed with other types of traffic, it represents an additional *constant* component to the amount of information received or transmitted in the given network node.

The packet size process presents the characteristics of a bimodal process, mainly influenced by the file download related traffic, and it is not modeled by any of the considered distributions. Instant messaging related traffic introduces a certain smoothing factor in the curves but, as they are dominated by the other types of traffic, it is not really noticeable. Weibull gets distinguished as being the best candidate for modeling the distribution of the interarrival times.

3.3.10 *Traffic Mixture 3: File Sharing, Streaming Download, and Instant Messaging.* The previous scenario was modified in order to accommodate a streaming download instead of a Web download. This simple change altered significantly the byte count distributions, which can now be modeled by Weibull and Normal distributions fairly (Normal for outgoing traffic and Weibull for the remaining data sets). All the remaining results described in the previous section remain the same.

3.3.11 *Traffic Mixture 4: VoIP (Skype), Streaming and File Sharing.* The traffic mixture containing Skype traffic was the one that surprised us the most. All the studied aspects of this traffic mixture can be modeled using the theoretical distributions considered herein. And this is valid for the packet size distribution as well. Skype encapsulates voice into variable length packets, with sizes varying between 100 and 500 bytes, resulting in highly dynamic traffic characteristics. The shape of the curve of the packet size distribution, however, is severely affected by the other types of traffic, being best approximated by the Pareto distribution. The Normal distribution prevails when it comes to approximate

the byte count distribution. The analysis of the autocorrelation returned values within the interval [0.322, 0.539], for the incoming traffic, [0.855, 0.889] for the outgoing traffic and [0.585, 0.729] for the complete communications trace. As previously, the Weibull distribution is the one that best describes the interarrival times process.

3.3.12 *Traffic Mixture 5: Web, Mail and Instant Messaging.* The trace containing the communications of a day in the life of a working station can be modeled by using Weibull distributions. Outgoing communications are positively correlated (values varying between 0.003 and 0.330), while the bit count per time unit of the incoming connections behaves randomly. We considered the case where a given person was using the computer to work, frequently browsing the web, receiving and sending mail and, here and there, using an instant messaging to chat with friends. The results obtained proved that Weibull is suitable for the characterisation of the interarrival times and of the byte count.

The irregularity of the packet size distribution led us once again to reject any of the theoretical models. Most of the incidences concerning this traffic aspect are concentrated around the typical maximum and minimum packet sizes for IP over Ethernet networks (in which the traces were collected). The asymmetry inherent to the client server paradigm is superbly emphasised in this P2P free scenario: small packets (corresponding mostly to HTTP requests and ACKs) occupy approximately 80% of the transmitted units, while 50% of the *received* packets is bigger than 1500 bytes.

4. SUMMARY AND DISCUSSION OF THE RESULTS

4.1 Summary

In Table VII, we summarize the results described in Section 3, presenting the best distribution for each case. The purpose is to provide a fast and easy way to read (and possibly use) the results of this work. As, for some cases, several distributions could fit the experimental distribution with similar accuracy, we chose to include only the distribution that has proven itself to be the best fit for the biggest number of other cases. In the Packet Size column, one will often find values and empirical frequencies that are not embraced by any of the notations introduced previously. They constitute the most probable values for the packet size found for the specific scenario, not the parameters of any of the considered models. For a more detailed discussion about the experimental distributions and the results of the fitting procedure, we refer the reader to Section 3.3.1.

4.2 Discussion

Table VII only puts emphasis on what was elaborated throughout the preceding sections. For the traffic aspects considered, Weibull and Normal distributions fit the majority of cases best. For this particular evaluation, the process describing the packet sizes is the one that behaves worst, being only possible to model using Weibull, Pareto, and Normal distributions when P2P traffic was present. In most of the cases, the packet size distribution is bimodal, presenting a probability concentration around two different values or small collections of values, the first around the maximum packet size and the other one around a smaller packet size, related to packets like acknowledgements or TCP SYN packets. The Weibull modeled the interarrival time distribution in almost all the cases. Even in the case (streaming broadcast-downstream) where Pareto was the most adequate distribution, Weibull fitted the experimental distribution well. It was not possible to fit a distribution for one case only, probably due to specific characteristics of the protocol.

The autocorrelation of almost all traces shows evidences of positive dependencies among the values of the bit count per second, at least until the time lag of 40s. It shows that the persistent properties of self-similar traffic are still embedded in the data when it is received at its destination, and that some of the traffic streams already exhibit positive correlations when first sent to the network. Persistence

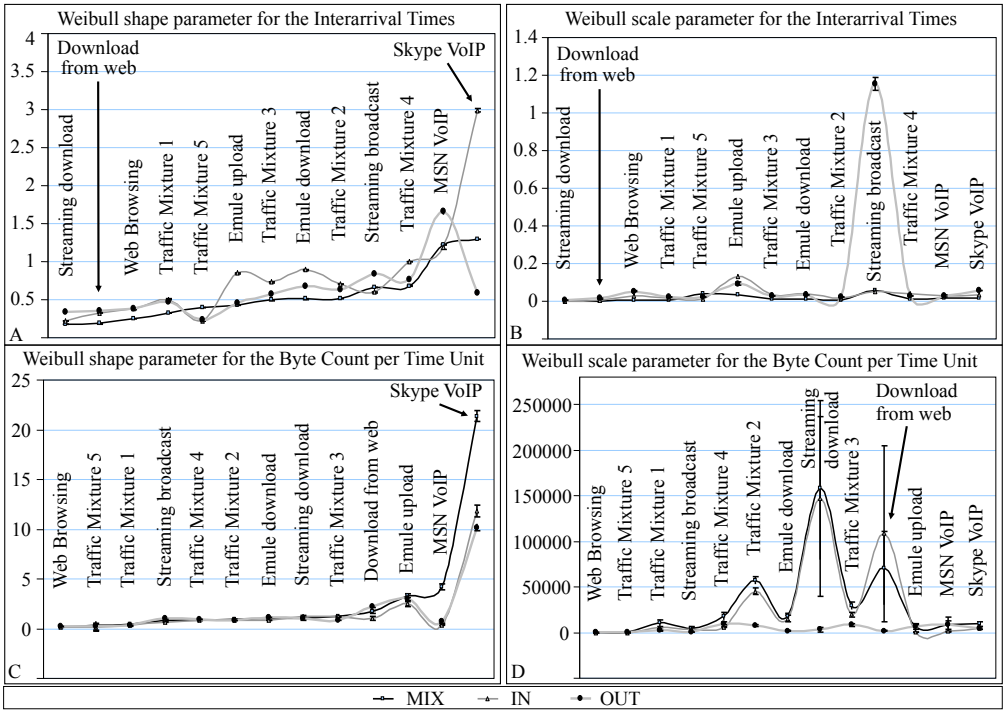

Fig. 5. The Weibull parameters plotted against the designation of each considered scenario: (a) the *shape* parameter values estimated for the *interarrival times*; (b) the *scale* parameter values estimated for the *interarrival times*; (c) the *shape* parameter values estimated for the *bit count per time unit*; (d) the *scale* parameter values estimated for the *bit count per time unit*. The several traffic classes are sorted in increasing order of the value of the scale parameter of the MIX data set.

is partially fomented by the type and increasing number of telematic applications and connections a single terminal machine handles nowadays. The correlation is then enhanced in network aggregation points, where the range of previously mentioned dependencies is further extended by the interaction between several traffic streams. In some cases, the terminal machine behaves like an aggregation point also, where several applications originate different subconnections that hustle to obtain the network resources.

4.3 The Distribution Parameters

To better depict how the specific characteristics of the several types of traffic are reflected in the distribution parameters, the estimated values of the Weibull distribution were plotted against the type of traffic and included in Figure 5. All the values are delimitated by Confidence Intervals (CIs) with 0.05 of significance, provided by the respective maximum likelihood estimator. Notice that some of CIs are too small to be observable in the charts, and that the order by which the applications appear in each chart was determined by the shape parameter of the MIX data set. The first two charts (Figure 5(a) and 5(b)) concern the analysis conducted for the Interarrival times process, while the two at the bottom (Figure 5(c) and 5(d)) are related to the analogous analysis of the byte count per time unit process.

In terms of interarrival times, the shape parameter of the Weibull distribution varies between 1.73E-1(± 2.11E-3) and 2.99(± 2.28E-2), as one moves from the streaming download to the Skype VoIP scenario (see Figure 5(a)). It is interesting to notice that the explicit difference between VoIP traffic and

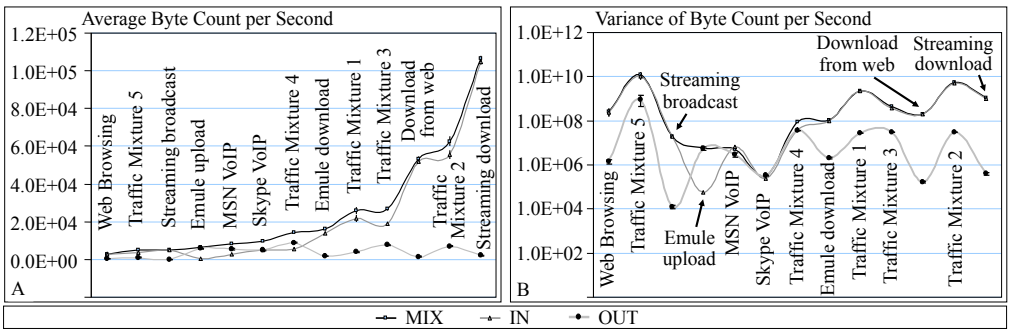

Fig. 6. The (a) average and the (b) variance of the byte count per time unit, plotted against the designation of each considered scenario. The several categories are sorted in increasing order of the average value of the MIX data set. (The y-axis of chart (b) is in logarithmic scale.)

the remaining types of traffic moves that kind of communications to the tail of the charts. As expected, the largest CIs occur when the respective data set is better described by a model different from Weibull. As may be seen in the chart Figure 5(b), the scale parameter of the referred distribution was low for most of the analysed traces, except for the *streaming broadcast*, whose largest value was of 1.15(± 3.62E-2), and for which no fit was found. Other examples of such behavior are found in Figures 5(c) and 5(d), where the estimates of the scale parameter for the byte count per second of the streaming download and of the download from web resent the fact that the referred process is better approximated by a Gaussian variable. For the IN datasets, the values of the scale parameter (and CIs) are 146843.49(± 11881.05) and 108510.43(± 72145.69), respectively.

Overall, the values of Figures 5(c) and 5(d) are higher than their counterparts in Figures 5(a) and 5(b), with the shape parameter varying between 2.42E-1(± 2.14E-2) (OUT dataset of Web Traffic without Streaming) and 21.38(± 5.55E-1) (MIX dataset of Skype VoIP). As opposed to the values obtained for the interarrival process, the scale parameter oscillates between a couple of hundreds to several hundred thousands, as a consequence of the magnitude of the metrics involved. Again, VoIP traffic is the one presenting the highest shape values, inspired by the small variance of the associated byte count per second and packet size distributions. The perfect symmetry of this P2P application is flagrant in the 4th and 5th lines of Table VII, being the network influences reflected in the empirical curves concerning the interarrival times.

Figures 6(a) and 6(b) contain the estimated values for the average and variance of the bit count per time unit, sorted in increasing order of the average value, and plotted against the designation of the scenario they refer to (note that, for reading commodity, the *y-axis* of the chart in Figure 6(b) was represented in logarithmic scale). By observing the plots, one may notice that streaming download was the one with the highest average (106554.18(± 1620.16)), reflection of a higher and steadier transmission debit (approximately 856Kbps). As a result of the asymmetry of the network communications at a terminal node, the outgoing traffic is, most of the times, characterized by a smaller transmission rate than the one of the incoming connections. The variance suffers inevitably from the same effect (the higher the debit, the more variable the bit rate is).

5. CONCLUSIONS AND FUTURE WORK

This article brings the focus of traffic analysis to the study of the behavior of network traffic sources. It provides a compressed version of the models that can be used to simulate the main traffic classes a source can produce, along with a compilation of the parameters that define them. After summarizing

the literature, we presented the results obtained from the analysis conducted over real traffic traces, collected in different scenarios and containing traffic from several classes, that characterize the traffic generated by an individual source (in the sense of end user or terminal traffic source). By fitting some well-known distributions to the experimental data, we tried to indicate the models that best describe the processes representing each traffic class. We have concluded that the Weibull distribution was capable of adapting itself to the majority of the empirical sequences of interarrival times and of bits per time unit. On the other hand, the distribution of the packet sizes is, most of the times, dominated by probability peaks around well defined values. It is thus suggested that the computational synthesis of such aspect should be made recurring either to previously collected data, or to the bi/trimodal distributions specified in this article (see Table VII).

It was also concluded that, for most of the analyzed traces the values of the bit count per time unit were positively correlated, at least upto the aggregation scale of 40s. This observation not only corroborates previous studies about the presence of self-similarity in network aggregation points, as it also suggests that one might want to simulate some dependence when modeling the aforementioned aspect of source traffic. The referred property achieves higher expression in scenarios where several streams are concurring for the same resource or large files are being exchanged, as is the case of file sharing traffic.

In the future, we plan to describe the impact of this work in simulations concerning prediction and trends of network traffic volume. It is also our intention to formalise the simulation procedure of the several traffic classes, and of their aggregation, in a separate publication. We found that the ACF of some of the traces could be satisfactorily approximated by a power law like the one in (3), which simplifies the simulation of that particular aspect, since it enables the usage of self-similar or multifractal series generators followed by their suitable transformation into the observed marginal distribution. That will be the subject of a more detailed research work, and of a future publication.

ACKNOWLEDGMENTS

The authors are thankful to all the anonymous reviewers who contributed constructively for the improvement of this work.

REFERENCES

ANSARI, N., LIU, H., SHI, Y. Q., AND ZHAO, H. 2002. On modeling MPEG video traffics. *IEEE Trans. Broadcast. 48,* 4, 337–347.

BARFORD, P. AND CROVELLA, M. 1998. Generating representative Web workloads for network and server performance evaluation. *ACM SIGMETRICS Perform. Evalu. Rev. 26,* 1, 151–160.

BERAN, J., SHERMAN, R., TAQQU, M. S., AND WILLINGER, W. 1995. Long-range dependence in variable bit-rate video traffic. *IEEE Trans. Comm. 43,* 234, 1566–1579.

BOLOTIN, V. A. 1994. Modeling call holding time distributions for CCS network design and performance analysis. *IEEE J. Selec. Areas Comm. 12,* 3, 433–438.

CAO, J., CLEVELAND, W. S., LIN, D., AND SUN, D. X. 2002. Internet traffic tends toward poisson and independent as the load increases. In *Nonlinear Estimation and Classification*, C. Holmes, D. Denison, M. Hansen, B. Yu, and B. Mallick, Eds. Springer, NY, 83–109.

CASILARI, E., CANO-GARCÍA, J. M., GONZÁLEZ-CAÑETE, F. J., AND SANDOVAL, F. 2004. Modelling of individual and aggregate Web traffic. In *High Speed Networks and Multimedia Communications*. Lecture Notes in Computer Science, vol. 3079. Springer, 84–95.

CASILARI, E., MONTES, H., AND SANDOVAL, F. 2002. Modelling of voice traffic over IP networks. In *Proceedings of the 3rd International Symposium On Communication Systems, Networks and Digital Signal Processing*. 411–414.

CISCO DOCUMENT SERVER. 2002. Traffic analysis for VoIP. Tech. rep. `http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/voipsol/ta_is%d.htm`.

COX, D. R. 1984. Long-range dependence: A review. In *Statistics: An Appraisal*. Iowa State University Press, 55–74.

CROVELLA, M. E. AND BESTAVROS, A. 1995. Explaining World Wide Web traffic self-similarity. Tech. rep. 1995-015, Computer Science Department, Boston University.

DAWOOD, A. M. AND GHANBARI, M. 1999. Content-based MPEG video traffic modelling. *IEEE Trans. Multimedia 1,* 1, 77–87.

FELDMANN, A., GILBERT, A. C., AND WILLINGER, W. 1998. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. *ACM SIGCOMM Comput. Comm. Rev. 28,* 4, 42–55.

FREEMAN, R. L. 2004. *Telecommunication System Engineering*, 4th Ed. Wiley-IEEE Press.

GARRETT, M. W. AND WILLINGER, W. 1994. Analysis, modeling and generation of self-similar VBR Video Traffic. *ACM SIGCOMM Comput. Comm. Rev. 24,* 4, 269–280.

HEYMAN, D. P. 1997. The GBAR source model for VBR videoconferences. *IEEE/ACM Trans. Netw. 5,* 4, 554–560.

HEYMAN, D. P., TABATABAI, A., AND LAKSHMAN, T. V. 1992. Statistical analysis and simulation study of video teleconference traffic in ATM networks. *IEEE Trans. Circ. Syst. Video Techn. 2,* 1, 49–59.

HUANG, C., DEVETSIKIOTIS, M., LAMBADARIS, I., AND KAYE, A. R. 1995. Modeling and simulation of self-similar variable bit rate compressed video: A unified approach. *ACM SIGCOMM Comput. Comm. Rev. 25,* 4, 114–125.

HUGHES, C. J., GHANBARI, M., AND PEARSON, D. E. 1993. Modeling and subjective assessment of cell discard in ATM video. *IEEE Trans. Image Process. 2,* 2, 212–222.

ISHAC, J. 2001. FTP traffic generator. Tech. rep., Case Western Reserve University.

JIANG, W. AND SCHULZRINNE, H. 2000. Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation. In *Proceedings of the 9th IEEE International Conference on Computer Communications and Networks*. 82–87.

KRUNZ, M. M. AND MAKOWSKI, A. M. 1998. Modeling video traffic using M/G/Inf input processes: A compromise between Markovian and LRD models. *IEEE J. Select. Areas Comm. 16,* 5, 733–748.

LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. 1994. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw. 2,* 1, 1–15.

LIU, H., ANSARI, N., AND SHI, Y. Q. 1999. Markov-modulated self-similar processes: MPEG coded video traffic modeler and synthesizer. In *Proceedings of The Global Telecommunications Conference (GLOBECOM'99)*. Vol. 2., 1184–1188.

LIU, X.-G. AND BABIARZ, J. 2007. Simulation results for explicit PCN marking and flow termination (Preemption). White paper, Nortel.

MAGLARIS, B., ANASTASSIOU, D., SEN, P., KARLSSON, G., AND ROBBINS, J. D. 1988. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. Comm. 36,* 7, 834–844.

MELAMED, B., RAYCHAUDHURI, D., SENGUPTA, B., AND ZDEPSKI, J. 1994. TES-based video source modeling for performance evaluation of integrated networks. *IEEE Trans. Comm. 42,* 10, 2773–2777.

PAXSON, V. AND FLOYD, S. 1995. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Trans. Netw. 3,* 3, 226–244.

RAMAMURTHY, G. AND SENGUPTA, B. 1992. Modeling and analysis of a variable bit rate video multiplexer. In *Proceedings of the INFOCOM Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 2. 817–827.

REININGER, D., MELAMED, B., RAYCHAUDHURI, D., AND SENGUPTA, B. 1994. Variable bit rate video: characteristics, modeling and multiplexing. In *Proceedings of the 14th International Teletraffic Congress*. Vol. 1a. 295–306.

ROSE, O. 1995a. Simple and efficient models for variable bit rate MPEG video traffic. Tech. rep., Institute of Computer Science, University of Wuerzburg.

ROSE, O. 1995b. Statistical properties of MPEG video traffic and their impact on traffic modelling in ATM systems. In *Proceedings of the 20th Conference on Local Computer Networks*. 397–406.

ROSE, O. AND FRATER, M. R. 1993. A comparison of models for VBR video traffic sources in B-ISDN. Tech. rep., University of Wuerzburg.

SEGER, J. 2003. Modelling approach for VoIP traffic aggregations for transferring tele-traffic trunks in QoS enabled IP-Backbone Environment. In *Proceedings of the 1st International Workshop on Inter-domain Performance and Simulation*.

SEN, P., MAGLARIS, B., RIKLI, N.-E., AND ANASTASSIOU, D. 1989. Models for packet switching of variable-bit-rate video sources. *IEEE J. Select. Areas Comm. 7,* 5, 865–869.

TCPDUMP. 2008. TCPDUMP public repository.

WILLINGER, W., PAXSON, V., AND TAQQU, M. S. 1998. Self-similarity and heavy tails: Structural modeling of network traffic. In *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, R. J. Adler, R. E. Feldman, and M. S. Taqqu Eds., Birkhuser, Boston, Chapter Applications, 27–53.

WILLINGER, W., TAQQU, M., SHERMAN, R., AND WILSON, D. 1997. Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level. *IEEE/ACM Tran. Netw. 5,* 1, 71–86.

Received February 2008; revised October 2008, February 2009, June 2009; accepted August 2009

Chapter 4

Exploring Behavioral Patterns Through Entropy in Multimedia Peer-to-Peer Traffic

This chapter consists of the following article:

Exploring Behavioral Patterns Through Entropy in Multimedia Peer-to-Peer Traffic

João V. Gomes, Pedro. R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

The Computer Journal (Oxford University Press), accepted for publication, 2011.

DOI: 10.1093/comjnl/bxr127

According to 2010 Journal Citation Reports published by Thomson Reuters in 2011, this journal scored ISI journal performance metrics as follows:

ISI Impact Factor (2010): 1.363

ISI Article Influence Score (2010): 0.592

Journal Ranking (2010): 15/48 (Computer Science, Hardware & Architecture)

Journal Ranking (2010): 54/128 (Computer Science, Information Systems)

Journal Ranking (2010): 30/99 (Computer Science, Software Engineering)

Journal Ranking (2010): 34/97 (Computer Science, Theory & Methods)

Exploring Behavioral Patterns Through Entropy in Multimedia Peer-to-Peer Traffic

João V. Gomes[1], Pedro R. M. Inácio[1], Manuela Pereira[1], Mário M. Freire[1] and Paulo P. Monteiro[2]

[1]*Instituto de Telecomunicações, Department of Computer Science, University of Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal*
[2]*University of Aveiro, Instituto de Telecomunicações, Nokia Siemens Networks, Rua Irmãos Siemens, 1, 2720-093 Amadora, Portugal*
Email: jgomes@penhas.di.ubi.pt, {inacio, mpereira, mario}@di.ubi.pt, paulo.1.monteiro@nsn.com

The inclusion of encryption or evasive techniques in popular applications increased the importance of characterizing network traffic based on behavior. This study aims to characterize peer-to-peer (P2P) traffic from the perspective of host computers by focusing on the packet lengths. The article explores the dissimilarities between the lengths of Internet Protocol (IP) packets generated by P2P and non-P2P applications. The heterogeneity of those lengths was assessed using entropy and compared for different classes of applications, through the implementation of a sliding analysis window. Initial observations show that the lengths of the packets generated by P2P applications are more varied than the ones of non-P2P applications. These patterns were used to implement a method to identify hosts running P2P applications. Unlike previous studies on this area, we used the heterogeneity of the packet lengths instead of the length value *per se* and a sliding window calculation procedure was adopted to allow real-time processing. The results of this study can be used for the characterization of traffic generated by P2P applications, as well as for traffic classification and management purposes.

Keywords: Internet Traffic; Multimedia Applications; Network Management; Peer-to-Peer Networks; Traffic Monitoring and Analysis

Received 00 July 2011; revised 00 Month Year

1. INTRODUCTION

The statistical analysis of different traffic parameters has always played an important role in the management of computer networks, since it enables the mathematical description of traffic behavior. The information retrieved is vital to achieve a better understanding of how the network resources are being used and how the traffic load impacts network capacity. In terms of network management, a good example of the importance of mathematically characterizing traffic is the well known study by Leland et al. [1]. With their work, the authors showed that a property known as self-similarity was embedded in the traffic, breaking a few old assumptions regarding the best probabilistic model to simulate or design the network.

The capability to mathematically characterize the traffic behavior, besides being used in the simulation of network traffic, has been employed mainly on the definition of strategies to effectively distribute the network workload and on the establishment of fair policies for bandwidth allocation, as well as on the design of more efficient networks. Network Intrusion Detection Systems (NIDSs), particularly the ones that elaborate on anomaly-based approaches, have also taken advantage of statistical analysis to describe the behavior of legitimate or illegitimate traffic, depending on whether they intend to identify the traffic that departs from a model of *normal* behavior or the traffic that matches a model of *abnormal* behavior [2].

More recently, new network paradigms, like peer-to-peer (P2P) computing, started to give users the power to act as content providers. This shift in the user role has modified properties of the traffic making it difficult to determine its nature [3]. From the network management perspective, P2P applications may raise a few challenges for network administrators, since the dual role (*client* and *server*) played by P2P hosts increase the traffic load in the edges. Furthermore, the

improvement of network throughput and the adoption of obfuscation techniques by popular applications have increased the computation power required to deeply analyze the data within the packets at high speeds and have reduced the effectiveness of the approaches that are based on the payload inspection [4]. Due to these reasons, statistical tools are also being used to profile the traffic, characterize its nature, or provide a classification based on the application protocol.

The study described in this article comprises the analysis of the traffic of several popular multimedia applications and protocols from a host-level perspective. Focusing the observation on the data transmitted by an end user helps to characterize the generic behavior that is inherent to the traffic generated by a specific application or class of applications. Additionally, it enables the understanding of the properties of the traffic generated by a single host. The importance of the host level characterization and the applicability of focusing on the role of each mode has already been shown in other studies [5, 6].

In a previous work [7], we collected network traffic at its source, i.e., near the terminal device generating the traffic (which embraces all the packets transmitted by a single user), analyzed different traffic properties, and described their behavior. Following that effort, the study presented herein aims to characterize the lengths of the Internet Protocol (IP) packets generated by P2P applications and compare them with the ones generated by *client-server* applications. Instead of relying only on the lengths of the packets, or on a range of lengths, this work uses their level of heterogeneity, which we measure by resorting the concept of entropy. Moreover, we propose a method that enables entropy evaluation in real-time. Most published studies on statistical traffic analysis used offline methods to process the data from complete flows [9]. In some cases, only the first packets or bytes in each flow were analyzed so that the traffic could be monitored in real-time [10]. In this work, we resorted to a method based on a sliding window with a constant size of N packets to assess the entropy of the traffic, from the beginning to the end of the capturing period in real-time. At each iteration of the sliding window, instead of recalculating the probabilities of all packet lengths within the window, we update the packet length probability for the packet that leaves the window and for the one that is added. After that, we calculate the entropy by updating the entropy obtained in the previous iteration of the sliding window.

The results of our analysis show that the lengths of the IP packets generated by P2P applications are more heterogeneous than the ones generated by the remaining applications. This behavior is clearly observable in the level of entropy obtained for each application, which is higher for P2P traffic. Using the entropy level, it is possible to distinguish the P2P Voice over Internet Protocol (VoIP) traffic, the P2P file-sharing and P2P video streaming traffic, and the non-P2P traffic for a host running a single application. In addition, we also studied the effect on entropy of running multiple simultaneous non-P2P applications in the same machine. Although, in these cases, the entropy level is still lower than its value for P2P traffic, it can can be similar in extreme situations. Nevertheless, the results obtained by studying only the outgoing portion of the traffic using a similar method show that it is possible to distinguish these cases. The ability to discriminate between P2P and non-P2P applications has motivated several studies [6, 8]. By recognizing the behavior of the traffic from generic P2P protocols, it is possible to identify P2P traffic even from new or unknown specific P2P applications.

The remainder of the article is structured as follows. Background concepts and related work are introduced before the analysis of the heterogeneity of the packet lengths. Afterwards, the evaluation of entropy and the obtained results are presented, followed by the description of a host-based classification scheme. The article finishes with a section devoted to conclusion.

2. EXPLORING TRAFFIC FEATURES

Several authors have been using features (or properties) extracted from packet fields or flow information (e.g., packet length, flow duration, addresses, ...), to mathematically describe the traffic from computer networks. Taken alone, or transformed using statistical functions, these features are employed as discriminators to characterize and isolate the behavior of specific network data [11], like traffic anomalies or application classes. Erman et al. [12], e.g., resorted to a set of 11 flow features (such as the total number of packets, the mean of the packet lengths, the total bytes, among others) and to a cluster algorithm to implement a traffic classifier. Freire et al. [13] analyzed the web page request length, the web response length, the inter-arrival time between requests, the number of requests per page, and the page retrieval time, and used the results to build a model for web traffic. Palmieri and Fiore [14] used the minimum, the mean, the maximum and the variance of the inter-arrival times and of the packet lengths to implement a traffic classification method. Tutsch et al. [9] used the self-similarity property to describe the traffic generated by a P2P file-sharing system. The technical report by Moore et al. [15] provides an extended list of discriminators to be applied on traffic characterization.

Some studies have already used the lengths of the IP packets as a feature for traffic characterization or classification [16, 17, 18]. Generally, the authors use the mean of the lengths, the total bytes per flow, the lengths of the first n packets, etc. Most of the times, they also combine them with other features, such as the inter-arrival times [19, 20]. In this article, instead of looking at the lengths individually, we focus on the relation between the different lengths. We analyze how varied

or homogeneous the packets lengths are for different classes of traffic. The traffic was processed using a sliding window with a constant size, which allow us to calculate entropy in real-time and assess the variations of the heterogeneity level iteratively across time. The real-time analysis of packet length heterogeneity and its quantification using entropy was used for the first time in an early version of the work presented herein [21]. Recently, Li et al. [22] used a similar approach to identify VoIP traffic. They calculated the entropy value offline for each complete flow, preventing its application to real-time traffic analyses.

Although several authors have already applied entropy in traffic studies, they used it to analyze other traffic features and for distinct purposes. The data generated by applications that encrypt the traffic is highly random. Therefore, entropy is sometimes used to reveal the randomness of the bytes within the payload of packets from encrypted traffic [23, 24]. Additionally, entropy has been widely use in the field of traffic anomaly detection. In most cases, authors rely on entropy to measure the randomness that some anomalies induce in the ports and IP addresses of the traffic from a host [25, 26, 27, 28].

Besides different traffic features, statistical analysis studies also use distinct observation levels. Most works analyze network traffic from the packet [29], flow [10], or session [30] perspectives. Generally, a flow is considered to include the packets transmitted between the same source and destination *address-port* pairs, in each or both directions (depending on whether one considers unidirectional or bidirectional flows), with a limited inter-arrival time and using the same transport protocol. On the other hand, a user session usually includes the packets of all flows concerning the communications of a certain application of a single user, until a predefined inactivity period is reached [30]. However, researchers are looking at the network data from different perspectives as a way to achieve an extended knowledge, improved by contrasting levels of views, about the traffic behavior. For instance, Khakpour and Liu [10] proposed the classification of the flows into *text*, *binary*, or *encrypted*, based on the nature of their contents. In this article, we focus on the host level. Such perspective enables the observation of the behavior of a single user and the characterization of his or her traffic. Moreover, it makes it possible to explore the role of a node in the network and the data as a combination of the impact of different applications from the same host.

3. METHOD FOR EVALUATING THE HETEROGENEITY OF THE PACKET LENGTHS

In order to analyze the heterogeneity of the packet lengths, we collected traffic from several multimedia applications, analyzed their level of heterogeneity, and compared it between P2P and non-P2P traffic. This section describes the research method used in this work and its implementation.

3.1. Experimental Network Data

The traffic traces used in this work were collected by tapping the connection of individual end users so as to obtain the traffic from a source level perspective. To be sure of what application has generated the analyzed traffic and avoid any case of misclassification, each user was running only one a application at a time. Hence, each trace corresponds to a single session of a certain application from one single user. Later on, we also captured traffic from individual users running several simultaneous non-P2P applications. These aggregated traces were used to evaluate the effect of multiple applications in a host traffic and how they impact the entropy results obtained for individual applications.

In this study, we chose services or applications that are widely used, are heavy bandwidth consumers, or raise more challenges from the perspective of traffic and network management, ending up with a set of applications with varied characteristics. Moreover, we tried to use applications with distinct behaviors to make it more challenging to identify common patterns.

The users were running distinct operating systems and each of them was connected in one of three different contexts: directly to the Internet, in a small Local Area Network (LAN), in a branch of a complex LAN. The following list describes the applications, protocols, or services whose traffic was analyzed:

- **Web browsing** – the activity of browsing web pages (excluding the streaming of multimedia contents, which is included in other categories);
- **Mail** – Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and Internet Message Access Protocol (IMAP);
- **Remote shell** – *Telnet* and Secure Shell (SSH);
- **File transfer** – File Transfer Protocol (FTP) and Secure File Transfer Protocol (SFTP);
- **Hypertext Transfer Protocol (HTTP) download** – the download of a non Hypertext Markup Language (HTML) file, e.g., an executable or a disc image;
- **Live audio streaming** – Microsoft Media Server (MMS), Real Time Streaming Protocol (RTSP), and *Flash*-based streaming;
- **On-demand audio streaming** – RTSP, HTTP, and *Flash*-based streaming;
- **Live video streaming** – MMS, RTSP, and *Flash*-based streaming;
- **On-demand video streaming** – RTSP, HTTP, and *Flash*-based streaming;
- **P2P video streaming** – *SopCast*, *PPStream*, and *TVU Player*;
- **P2P file-sharing** – *BitTorrent*, *eDonkey*, and *Gnutella*;

- **P2P VoIP** – *Skype*, *Google Talk*, and *MSN* Messenger.

The use of long traces is usually a requirement to conduct sound studies on network traffic. However, given the particular nature, purpose, and approach of this work, it would be unrealistic to collect very long datasets of some kinds of traffic, e.g., VoIP or mail. Instead, we decided to analyze several traces of each class, with realistic durations, and verify the consistency of the results for all of them. The traces include 11.7 GB of data. The sum of the duration of all the datasets is approximately 61 hours. The datasets were collected in different periods, from October 2006 to September 2011. The traces containing aggregated data from multiple applications sum up to 13.3 GB and correspond to a 24 hours period.

In order to keep the article short, the figures included in the following sections contain examples for specific classes of traffic that are demonstrative of the conclusions obtained for all the datasets used in this study. Each chart depicts a single and entire trace as an example of the corresponding application. We intentionally tried to vary the examples shown in each of the figures.

3.2. Lengths of the Packets from P2P and Non-P2P Traffic

The traffic from non-P2P applications is generally formed by *well behaved* flows with a predictable and stable nature, most of the times based on a single or a few traditional *client-server* connections. On the other hand, P2P applications present a more chaotic nature, in terms of connections, that derives partially from the dual role played by a P2P user (which acts both as client and server). Moreover, especially in the case of file-sharing and video services, the P2P applications receive and provide data from and to more than one peer, generating multiple concurrent connections, possibly with distinct properties.

When observing the traffic from a P2P node, this chaotic nature is revealed in different aspects: the number of destination ports to which a single source port communicates, the multiple concurrent connections, the use of both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) between the same peers, etc [8]. These properties produce a measurable influence in the characteristics of the packet lengths.

From the datasets captured, we selected only the IP packets and excluded the packets of Internet Control Message Protocol (ICMP), Domain Name System (DNS), and local communications (packets exchanged with routers, gateways, and other local devices). Figure 1 depicts the distribution of the lengths over time, for six examples of traffic. The lengths of the packets transmitted by the P2P applications are extremely heterogeneous when compared to the ones from non-P2P applications. Figure 2 includes the cumulative probability distributions of the packet lengths so as to make it easier to understand the diversity and the probability weight of the values in the examples depicted in Figure 1. The discrepancy in the number of connections, between the different classes of applications, is then illustrated by Figure 3. The diameter of the circles, which is proportional to the number of packets transmitted, gives an idea of how the packets are distributed by multiple connections between different hosts.

The traffic from the HTTP download is composed, almost exclusively, by large packets, limited by the Ethernet Maximum Transmission Unit (MTU) (1500 bytes of payload plus the size of the header), and by small packets used to send TCP acknowledgement messages. The intermediate values that are visible in the streaming examples result from the aggregation of the application protocol units and its distribution into a few transport datagrams. The small and large values, as well as the intermediate value for live streaming, are clearly observable in the cumulative functions, where the completely horizontal lines between the probability jumps show their predominance.

In the case of P2P file-sharing and P2P video streaming, a host establishes several parallel connections with other peers. These connections may follow different paths with distinct MTUs, imposing different maximum limits for the packet lengths. This behavior is observable in Figure 1, in the charts related to these types of traffic, where it is possible to see a few horizontal lines formed by the packet lengths of connections with distinct MTUs. The biggest difference to non-P2P traffic, though, is the diversity of packets with distinct lengths used to establish and control the connections between the peers, and, mostly, to search for peers with the required contents and answer to requests from other users. Although the corresponding cumulative probability distributions, in Figure 2, still present a few probability jumps, they are smaller and less expressive when compared to non-P2P traffic. Moreover, the probabilities between the peaks increase smoothly. This effect is caused by the larger diversity of packet lengths which reduces the highest probabilities and increases the lowest ones.

Like the other P2P applications, P2P VoIP also generates traffic formed by heterogeneous packet lengths. Nevertheless, VoIP applications do not establish a large number of connections. The packet length heterogeneity is caused by the real-time requirements inherent to any phone call. Instead of transmitting the data in large packets limited by the MTU, the voice is codified in small blocks with different lengths that are sent immediately. This results in packets with extremely varied lengths concentrated in a very short range. This behavior is observable in Figure 2, where the cumulative probability increases almost uniformly from 0 to 1 as the packet lengths

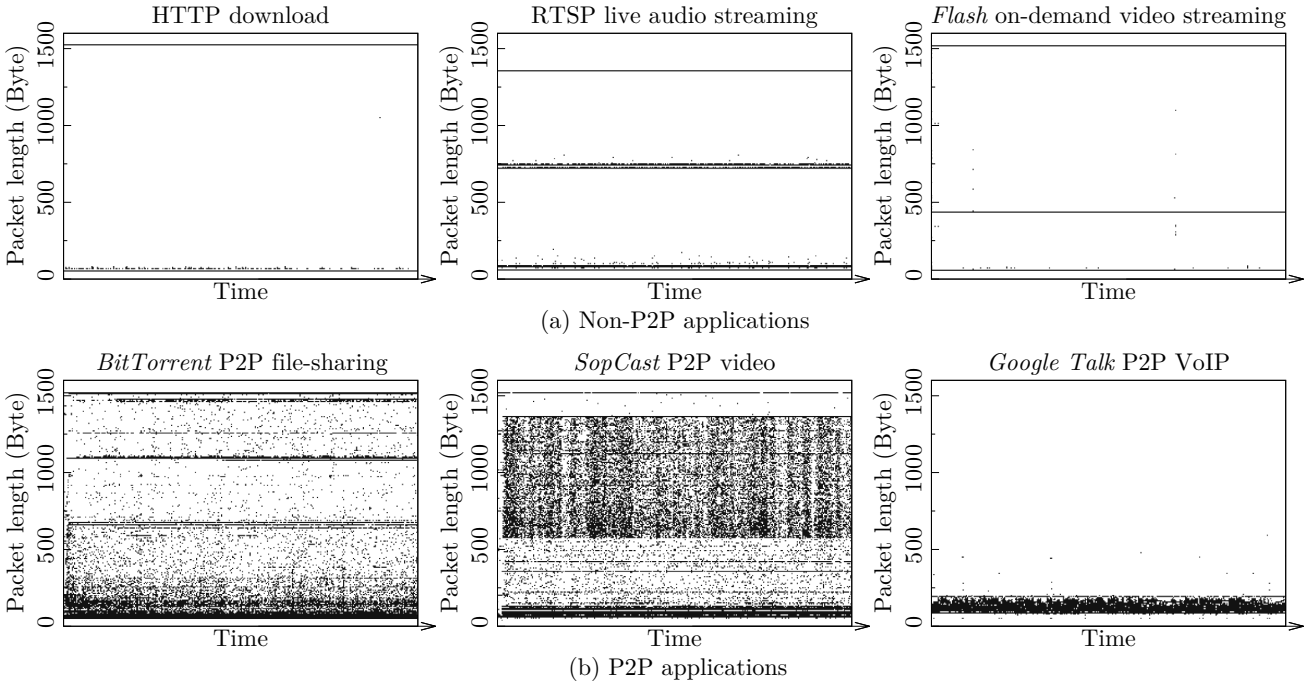

FIGURE 1. Distribution of the packet lengths versus time for different examples of traffic from (a) non-P2P and (b) P2P applications.

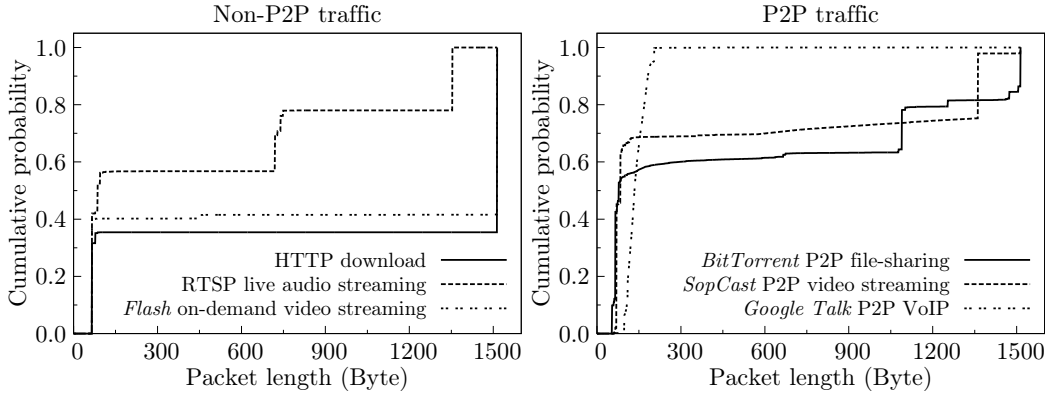

FIGURE 2. Cumulative probability distributions of the packet lengths.

vary between 100 and 200 bytes. Although some non-P2P services, like live streaming, also have real-time requirements, they resort to a buffer to aggregate data into larger blocks and still provide the contents smoothly.

3.3. Evaluation of the Heterogeneity of the Packet Lengths

In order to quantify the level of heterogeneity of the packet lengths described above, we employed entropy. This subsection explains the concept used in this work and presents how the entropy calculation method was implemented in our experiments.

3.3.1. Evaluation of the Heterogeneity Using Entropy
In this article, we use entropy as a measure to express the level of heterogeneity of the lengths of IP packets. The concept of entropy used herein coincides with the one introduced by Shannon in the information theory [31], where entropy is presented as a measure of the uncertainty of a random variate.

Entropy is frequently denoted by $H(x)$ and defined by an expression equivalent to (1), where n represents the number of values of x for which the statistic is calculated, and $p(x_i)$ is the probability of the particular value of x_i:

$$H(x) = -\sum_{i=1}^{n} p(x_i) \ln p(x_i). \qquad (1)$$

For any finite number $n \in \mathbb{N}$, the maximum value that

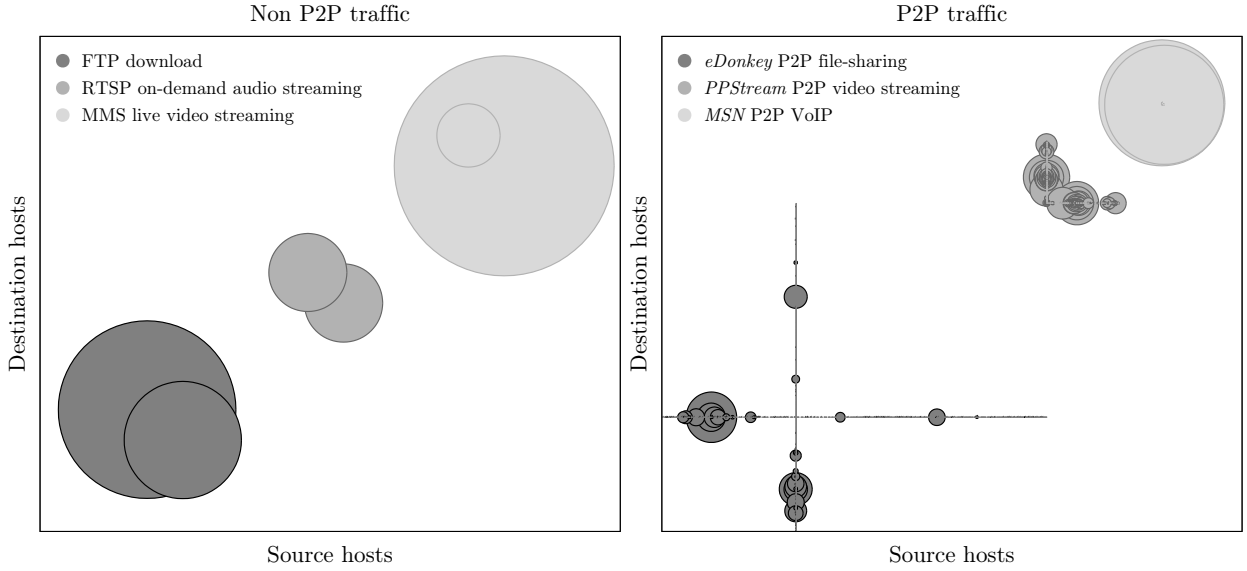

FIGURE 3. Schematic representation of the number of packets transmitted between the same source and destination addresses, in a single user session, for non-P2P and P2P traffic.

$H(x)$ may reach is given by

$$H(x) = \ln n. \tag{2}$$

The value of entropy is always a positive number. If an estimate of $H(x)$ is close to 0, the number of different values in the pool of samples is small. $H(x)$ increases with the number of different occurrences under analysis.

3.3.2. Applying Entropy to the Lengths of the Packets Using the level of heterogeneity of the packet lengths as a factor for traffic characterization requires the ability to quantify and describe it mathematically. In order to do so, we resort to the definition of entropy given by Shannon. The only thing we had to take into consideration was the fact that, as any other statistic and due to the *Law of Large Numbers*, the entropy value converges to a fixed value for an increasing pool of values. Therefore, we calculated entropy for a sliding window of N packets. In each step, the oldest value (i.e., the packet length) leaves the window and a new one is added, producing one entropy value for each iteration. The use of a sliding window provides the ability to analyze the traffic in real-time.

In order to implement the calculation method in an efficient way, we avoid recalculating the entropy for all packets within the sliding window in every iteration. Instead, we calculate the entropy using (1) when the window is filled for the first time. After that, in each iteration i of the sliding window, we:

1. subtract, to the entropy in the previous iteration, the weight of the oldest packet length (l_o) in the sliding window:
   $-(-p_{i-1}(l_o)\ln p_{i-1}(l_o))$;
2. subtract, to the result of step 1, the weight of the latest arrived packet length (l_l), in case at least one occurrence of l_l already exists in the sliding window:
   $-(-p_{i-1}(l_l)\ln p_{i-1}(l_l))$;
3. update the probabilities of l_o and l_l in iteration i after l_o leaves the window and l_l is added;
4. add, to the result of step 2, the weight of l_o in iteration i of the sliding window:
   $+(-p_i(l_o)\ln p_i(l_o))$;
5. add, to the result of step 4, the weight of l_l in iteration i of the sliding window:
   $+(-p_i(l_l)\ln p_i(l_l))$.

We aggregate these operations in (3) and we update the entropy in each iteration i of the sliding window using

$$U(l) = p_{i-1}(l)\ln p_{i-1}(l) - p_i(l)\ln p_i(l), \tag{3}$$

$$H_i(x) = H_{i-1}(x) + U(l_o) + U(l_l). \tag{4}$$

4. RESULTS AND ENTROPY ANALYSIS

The approach described in the previous section was used to analyze the entropy for all the captured datasets. In this section, we present the results obtained for the different classes of applications. Furthermore, we describe a scheme to distinguish between P2P and non-P2P traffic.

4.1. Analysis of Traffic Entropy at the Host Level

The results obtained in the entropy analysis were consistent with what was observed in the previous subsection. The chaotic behavior described above is reflected in the entropy level, which is clearly distinct for P2P and non-P2P traffic. The non-P2P datasets originate lower entropy values when compared

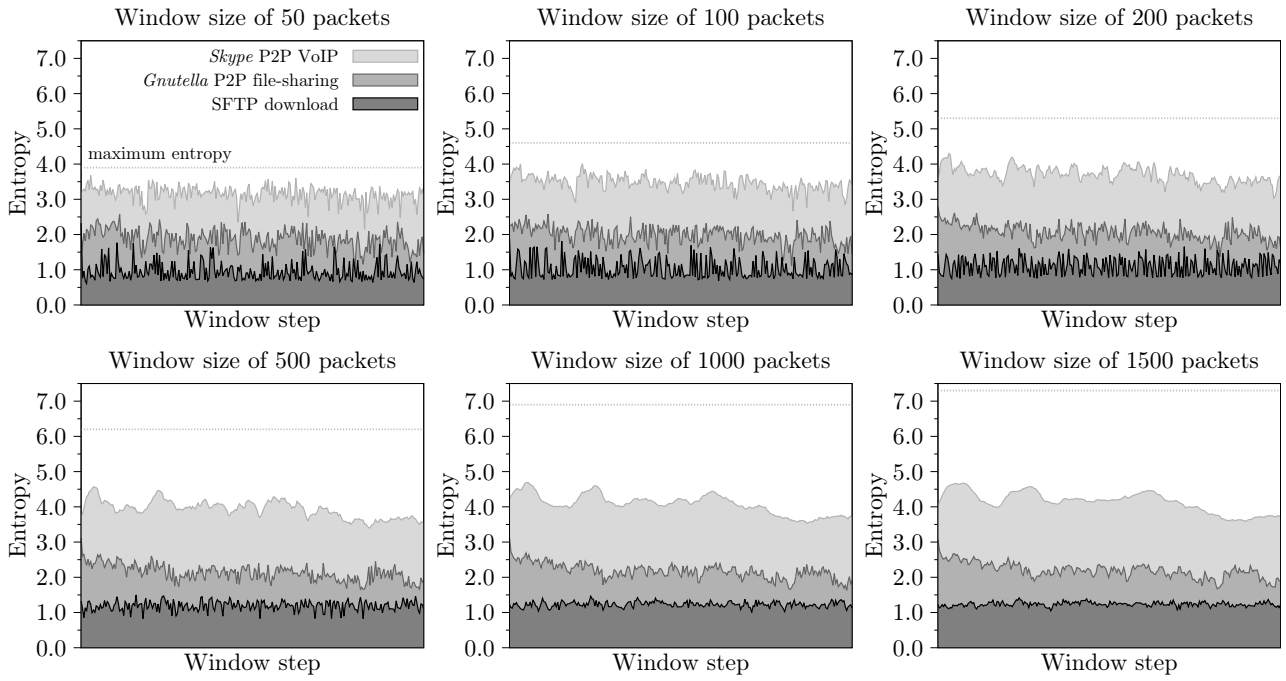

FIGURE 4. Evolution of the entropy value for sliding windows with constant sizes.

to what happens with the P2P traffic. In fact, the differences are also visible between VoIP and the remaining P2P applications. The entropy values for P2P video streaming and P2P file-sharing traffic fall into an intermediate level, between the ones for non-P2P applications and for VoIP data, whose entropy is close to the maximum value. Figure 4 shows three examples of this analysis and provides a graphical perspective of this behavior. It is possible to see that the entropy values for the *Gnutella* dataset is always between the values obtained for SFTP and *Skype* traffic.

In order to provide a more general and integrated perspective of the results obtained for all the datasets analyzed, we organized the entropy values in Table 1, sorted in descending order. For each trace we captured, we obtained the mean of the entropy in all the steps of the sliding window, ending up with one representative value per trace. After that, we calculated the mean for all the datasets from the same application. The rows in the table refer to the global mean of the entropy of all the traces of the related application or service. As it is possible to observe, almost all the P2P applications (which correspond to the highlighted rows) were ranked in the top of the table, with the VoIP examples taking the first places and having clearly higher entropy values than all the other applications.

In a TCP connection, some packets serve the sole purposes of starting, finishing, or managing the connection and, therefore, do not contain any application data. This fact is naturally inherent to the protocol and will be the same no matter what kind of traffic (P2P or non-P2P) will be transported by the TCP connection. Since we are focusing our attention in the dissimilarities of the traffic from different applications, we tried to minimize any common points that do not result from the application itself. Hence, we repeated the same trace analysis but, this time, we excluded all the TCP packets that are only used to send TCP flags and do not contain any transport payload. The results included in the right side of Table 1 show only a slight increase of the entropy of the most entropic datasets. However, the entropy of the traces whose packets have more homogeneous lengths decreased considerably to values near zero, emphasizing the entropy differences to the datasets containing packets with more heterogeneous lengths.

The window size impacts the entropy value. Therefore, we conducted this analysis for several distinct window sizes from 10 to 2000 packets. In Figure 4, we included examples for windows with 50, 100, 200, 500, 1000, and 1500 packets. In all the charts depicted in the figure, an horizontal line indicates the maximum value entropy can reach for the size of the window. For the same data, entropy increases slightly when the size of the window is bigger, as it can be seen not only in the figure, but also in Table 1. Nonetheless, this behavior is hardly noticeable for windows larger than 500 packets. In fact, the most relevant consequence of increasing the size of the window is the decrease of the entropy peaks and the magnification of the lower values, which ends up creating a smoothing effect that makes the entropy value more stable. The choice of an optimal window size will depend on the level of granularity required for

TABLE 1. Mean of the entropy of all the datasets of each application for sliding windows with sizes of 100 and 500 packets.

All the packets				**Excluding packets with no TCP payload**			
100 Packets Window		500 Packets Window		100 Packets Window		500 Packets Window	
Service or application	Entropy	Service or application	Entropy	Service or application	Entropy	Service or application	Entropy
Skype VoIP	3.689092	*Skype* VoIP	4.253871	*Skype* VoIP	3.692575	*Skype* VoIP	4.253475
Google Talk VoIP	3.673437	*Google Talk* VoIP	4.223415	*Google Talk* VoIP	3.672914	*Google Talk* VoIP	4.222280
MSN VoIP	3.304374	*MSN* VoIP	3.767171	*MSN* VoIP	3.306609	*MSN* VoIP	3.765748
eDonkey file-sharing	2.326302	*BitTorrent* file-sharing	2.600922	*BitTorrent* file-sharing	2.372140	*BitTorrent* file-sharing	2.597633
BitTorrent file-sharing	2.225038	*eDonkey* file-sharing	2.563792	*eDonkey* file-sharing	2.315821	*eDonkey* file-sharing	2.590145
Gnutella file-sharing	1.735907	*Gnutella* file-sharing	1.814077	eMail IMAP	1.704473	eMail IMAP	1.886539
eMail IMAP	1.636527	*SopCast* video	1.788471	*TVU* video	1.557982	*SopCast* video	1.712150
SopCast video	1.615137	eMail IMAP	1.765541	*SopCast* video	1.548545	*TVU* video	1.711503
TVU video	1.568804	*TVU* video	1.724189	*Gnutella* file-sharing	1.508016	*Gnutella* file-sharing	1.627393
Telnet session	1.547972	*Telnet* session	1.665598	*Telnet* session	1.344495	Web browsing	1.538494
Web browsing	1.35992	Web browsing	1.616038	*PPStream* video	1.317305	*Telnet* session	1.488726
RTSP live audio	1.40257	*PPStream* video	1.481672	RTSP live audio	1.276242	*PPStream* video	1.477753
PPStream video	1.320381	RTSP live audio	1.430851	Web browsing	1.183305	RTSP live audio	1.302925
HTTP on-demand audio	1.293733	*Flash* live audio	1.422119	RTSP on-demand audio	1.152338	HTTP on-demand audio	1.241080
Flash live audio	1.273091	HTTP on-demand audio	1.409538	HTTP on-demand audio	1.103575	RTSP on-demand audio	1.188753
RTSP on-demand audio	1.259748	RTSP on-demand video	1.305689	eMail SMTP	0.984063	*Flash* live audio	1.125196
RTSP on-demand video	1.150866	RTSP on-demand audio	1.289097	*Flash* live audio	0.938838	RTSP on-demand video	1.038351
MMS live audio	1.142598	*Flash* live video	1.268098	RTSP on-demand video	0.813354	eMail SMTP	0.964299
Flash live video	1.124564	SFTP download	1.190365	*Flash* live video	0.709199	*Flash* live video	0.904862
SSH session	1.10337	SSH session	1.159284	SSH session	0.671425	SSH session	0.744112
SFTP download	1.017874	MMS live audio	1.156805	MMS live audio	0.660225	MMS live audio	0.670584
eMail SMTP	0.989311	RTSP live video	1.015639	RTSP live video	0.481099	RTSP live video	0.620572
RTSP live video	0.889848	eMail SMTP	0.967104	SFTP download	0.331281	SFTP download	0.358668
HTTP download	0.865411	*Flash* on-demand video	0.831476	*Flash* on-demand audio	0.176783	MMS live video	0.180871
Flash on-demand video	0.795265	HTTP on-demand video	0.782923	MMS live video	0.167241	*Flash* on-demand audio	0.179289
Flash on-demand audio	0.770837	*Flash* on-demand audio	0.775613	HTTP download	0.147586	*Flash* on-demand video	0.150160
HTTP on-demand video	0.754165	HTTP download	0.701071	*Flash* on-demand video	0.125844	HTTP download	0.118539
eMail POP	0.71003	eMail POP	0.695409	eMail POP	0.071632	eMail POP	0.070216
FTP download	0.672094	FTP download	0.672138	HTTP on-demand video	0.054755	HTTP on-demand video	0.065906
MMS live video	0.63307	MMS live video	0.667665	FTP download	0.000054	FTP download	0.000069

a specific purpose. In the additional analyses described in the following subsections, we used sliding windows with size of 100 packets, as a compromise between the stability of the entropy level and the level of detail needed for the analysis.

4.2. Entropy of Simultaneous Applications

The analysis of the traffic from a host level perspective is straightforward when there is only a single application transmitting data. However, a regular user is running, most of the times and simultaneously, several applications. Therefore, following the same controlled approach used to collect the traces studied in the previous subsection, we captured a few datasets from an end-user running several P2P and non-P2P applications at the same time.

4.2.1. Aggregated Traffic from P2P and Non-P2P Applications

Firstly, we wanted to analyze the effect of non-P2P traffic in the entropy level when mixed with traffic from P2P applications. In the results we obtained, the presence of P2P traffic is still noticeable when the dataset contains traffic from more than one application. The weight of each application in the overall entropy value depends also on the amount of data transmitted by each of them. Because of their greedy nature in terms of bandwidth consumption, P2P applications tend to have a strong influence on the entropy of the trace.

Figure 5 presents two examples of this analysis. In order to maximize the possible variations of the entropy value, we chose to exclude all the TCP packets without payload from the analysis of these composed traces. In the first one, it is possible to observe that entropy increases as we add more P2P applications, especially when the VoIP call starts. In the end of the trace, when there is only the HTTP download running, entropy falls to values close to zero. After a human verification of the traffic in this period, we found that the small peaks in the entropy values are caused by Distributed Hash Table (DHT) systems implemented by most *BitTorrent* clients. These distributed systems are used to improve the search for contents and reduce the dependency on centralized trackers. They enable the content sharing when the tracker is not available or between peers that are not connected to the same tracker. In the datasets, it was possible to observe that, even after the application is stopped, the host was still receiving requests through the DHT system.

In the second example (bottom chart of Figure 5), a similar behavior is observable. Although *Google Talk* traffic raises entropy to high values, the larger share of *SopCast* data in the overall traffic pulls the entropy value down to a level more alike to what is common for P2P file-sharing and P2P video applications. It is only when the *SopCast* is stopped that the entropy values increase to a level similar to what we would expect for VoIP traffic.

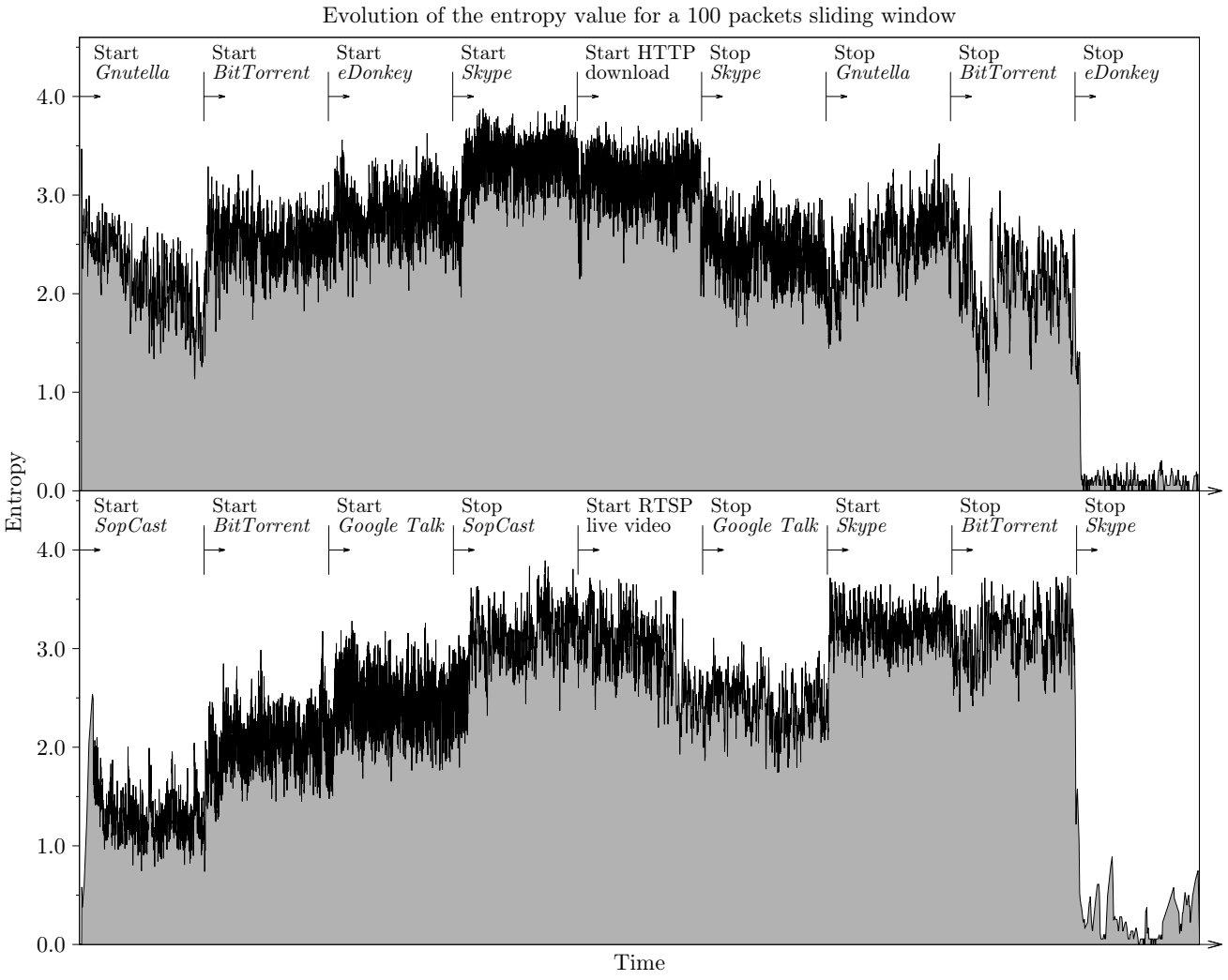

FIGURE 5. Entropy for two examples of datasets containing aggregated traffic from several simultaneous applications.

4.2.2. Aggregated Traffic from Several Non-P2P Applications

The heterogeneity of the packet lengths resulting from P2P applications is still noticeable if the same host is also running non-P2P applications. A distinct problem, though, is the possible effect in the entropy level of running several non-P2P applications in the same machine. In the case of P2P VoIP applications, the entropy is high for the single flow used to transmit the session data. However, as described above, for P2P file-sharing and P2P video streaming, the entropy is high due to the aggregation of multiple flows with different properties. Therefore, we wanted to analyze if the aggregation of traffic from multiple non-P2P applications could also result in an entropy level similar to the one obtained for P2P traffic, making it difficult to distinguish between both classes of traffic.

In order to perform such analysis, we collected traffic from single hosts running several non-P2P applications. We first captured a set of traces from hosts listening a few audio streaming sessions using different protocols, browsing the web, downloading a large file using HTTP, and checking the email. The traces of non-P2P traffic included in this set will be named as *type 1* in this article. In a second stage, we wanted to exaggerate the analysis, even if it could be unrealistic. Hence, we captured a different set of traces from hosts running many audio and video streaming sessions, besides of the applications used in the traces from type 1. These traces will be named, herein, as *type 2*.

Figure 6 compares the results obtained for a trace example containing P2P traffic, a type 1 trace, and a type 2 trace. In order to make a stricter comparison, we chose a P2P trace with an entropy level closer to the entropy values obtained for the non-P2P applications. The dark line depicts the mean so as to make it easier to compare the results. The entropy for the type 1 trace is very close to 1, being difficult to be distinguished from the P2P traffic. Nevertheless, as it was seen in Table 1, if the TCP packets with no payload are excluded from the analysis, the difference between the type 1 trace and P2P traffic is clear. However, the

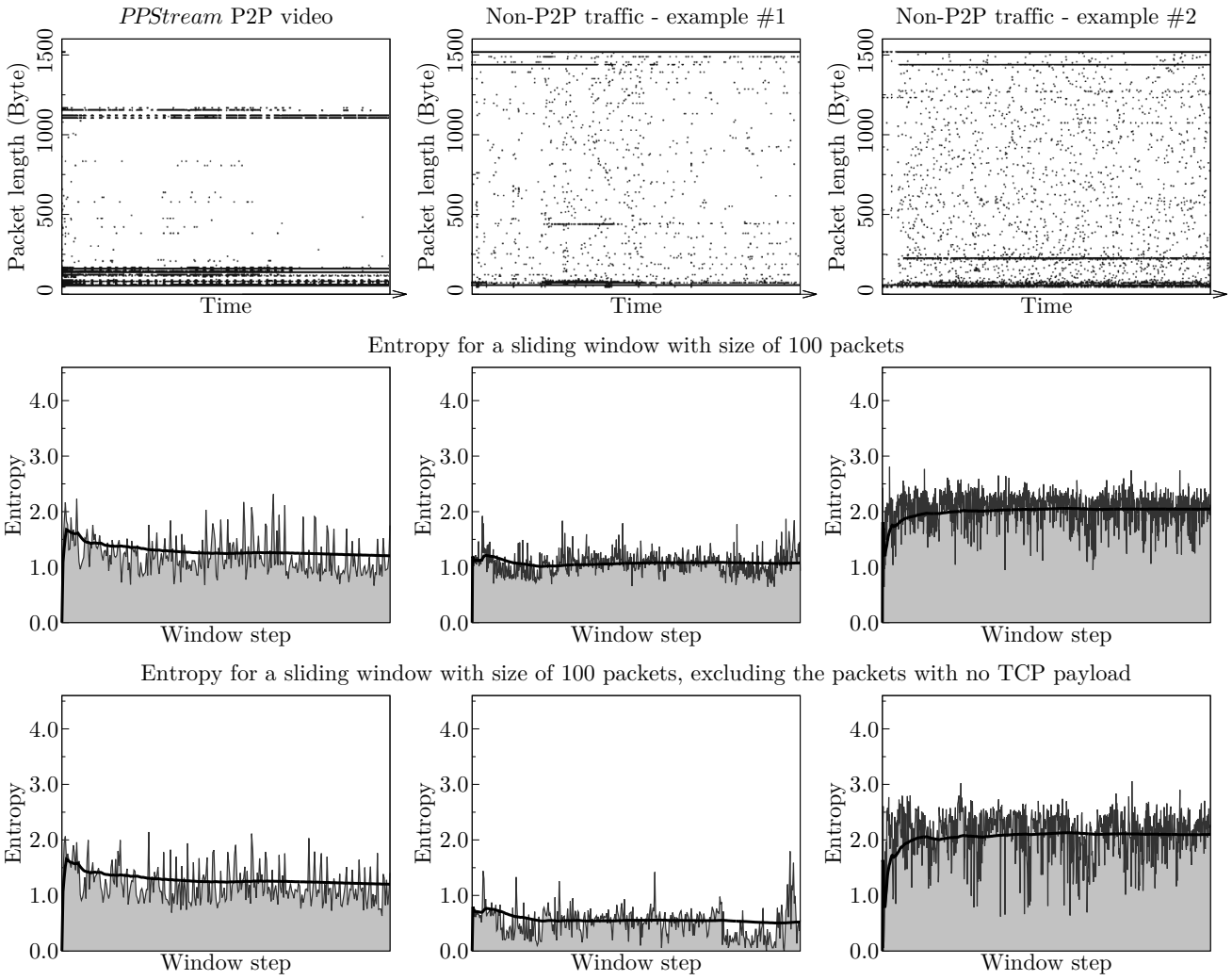

FIGURE 6. Packet lengths and the corresponding entropy for examples of P2P traffic and type 1 and type 2 traces of aggregated traffic from several simultaneous non-P2P applications.

type 2 trace presents an entropy level similar to the patterns evidenced by P2P traffic, even if the packets with no TCP payload are excluded from the analysis. Thus, in some cases, considering extreme examples, further analysis is required so as to make it possible to distinguish P2P traffic.

Since P2P hosts also act as servers, it is reasonable to think that the outgoing traffic may have different properties when compared with non-P2P applications. Hence, we focused the analysis in the outgoing traffic. Figure 7 presents a comparison of the entropy level of the outgoing traffic for the same three examples. One may observe a small difference between P2P traffic and the type 1 trace. However, the type 2 trace continues to present a behavior very similar to P2P traffic. As the lengths of the packets from non-P2P traffic are usually within a shorter range of values, we performed a different analysis by considering slots of lengths instead of each length individually. This analysis was implemented in a simple way by calculating the integer division of each length by the size of the slot we wanted to consider and then calculating entropy for the resultant values. For example, for slots of 200 bytes, every length from 0 to 199 bytes would be transformed into 0. We performed this analysis using slots with different sizes and we achieved better results with slots of 200 bytes. The results for the three examples are depicted in the third row of charts in Figure 7. In this case, the entropy values will be lower as there are less distinct possible lengths. For this reason, in these examples, the y axes range from 0 to 1. The entropy level for the aggregated non-P2P traffic is lower than for the P2P trace. These patterns are not so visible in the beginning of the traces, as that is the period where the requests are being made and no contents are being transmitted yet.

Similarly to the first table, Table 2 summarizes the results of this analysis. Each value represents the entropy mean for all the traces of the same application or class. The table presents the results obtained for

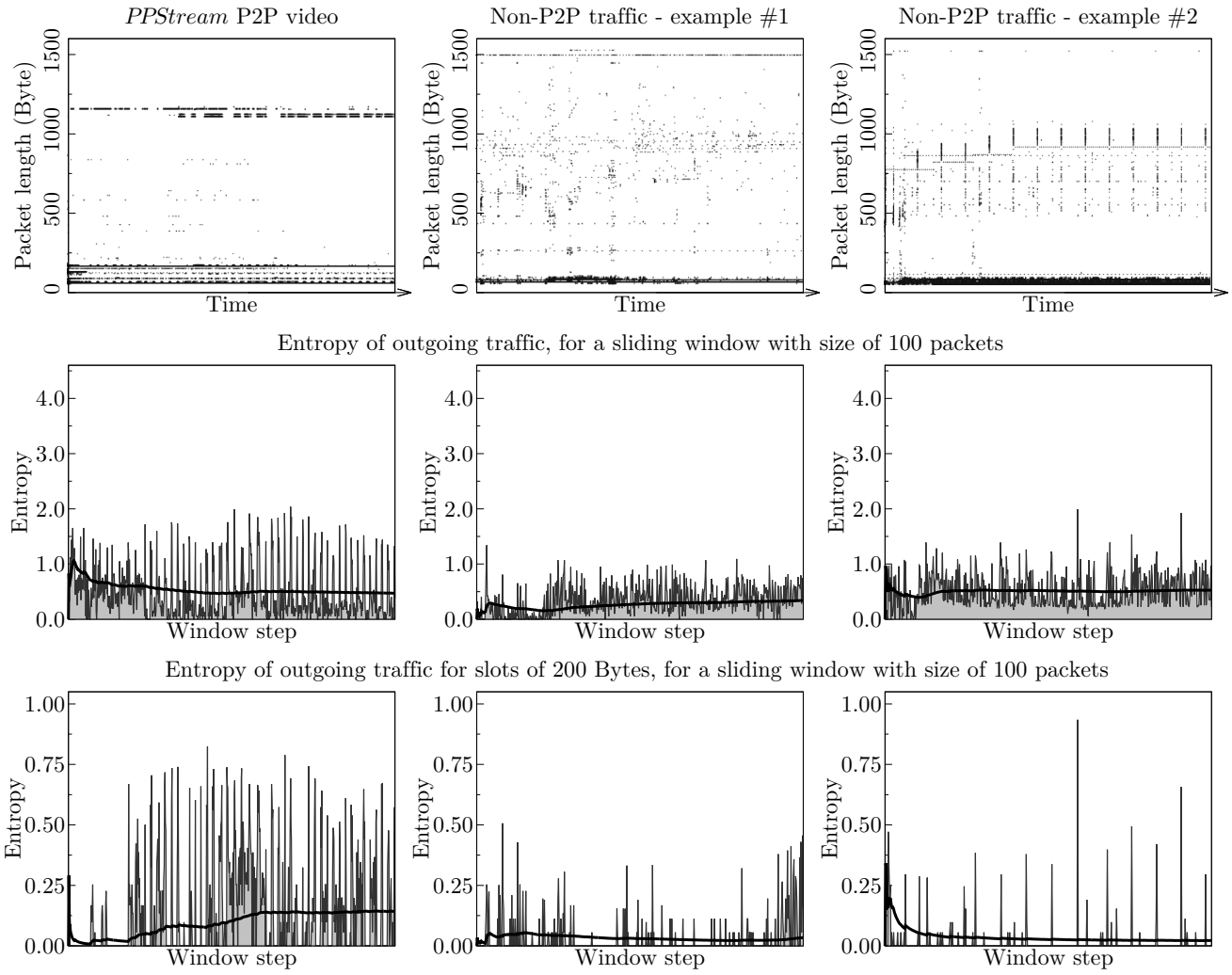

FIGURE 7. Packet lengths and the corresponding entropy of the outgoing traffic for examples of P2P traffic and type 1 and type 2 traces of aggregated traffic from several simultaneous non-P2P applications.

TABLE 2. Mean of the entropy for all the datasets of each class of application for sliding windows with size of 100 packets.

All packets		**Excluding packets with no TCP payload**		**Outgoing traffic**		**Outgoing traffic considering slots of 200 Bytes**	
Service or application	Entropy	Service or application	Entropy	Service or application	Entropy	Service or application	Entropy
eDonkey file-sharing	2.326302	*BitTorrent* file-sharing	2.372140	*eDonkey* file-sharing	1.803879	*eDonkey* file-sharing	0.697760
BitTorrent file-sharing	2.225038	*eDonkey* file-sharing	2.315821	*BitTorrent* file-sharing	1.564682	*BitTorrent* file-sharing	0.479357
Non-P2P traces type 2	1.881326	Non-P2P traces type 2	1.859609	*TVU* video	1.519015	*SopCast* video	0.434564
Gnutella file-sharing	1.735907	eMail IMAP	1.704473	*SopCast* video	1.408216	Web browsing	0.406709
eMail IMAP	1.636527	*TVU* video	1.557982	eMail IMAP	1.368927	*TVU* video	0.347328
SopCast video	1.615137	*SopCast* video	1.548545	*Gnutella* file-sharing	0.982682	*PPStream* video	0.200514
TVU video	1.568804	*Gnutella* file-sharing	1.508016	Telnet session	0.910888	*Gnutella* file-sharing	0.124701
Telnet session	1.547972	Telnet session	1.344495	Web browsing	0.910178	Non-P2P traces type 1	0.035944
RTSP live audio	1.40257	*PPStream* video	1.317305	*PPStream* video	0.675948	Non-P2P traces type 2	0.030780
Web browsing	1.35992	RTSP live audio	1.276242	Non-P2P traces type 2	0.522932	eMail IMAP	0.019976
PPStream video	1.320381	Web browsing	1.183305	Non-P2P traces type 1	0.413866	RTSP on-demand audio	0.005974
HTTP on-demand audio	1.293733	RTSP on-demand audio	1.152338	HTTP on-demand audio	0.177408	HTTP on-demand audio	0.000920
RTSP on-demand audio	1.259748	HTTP on-demand audio	1.103575	RTSP live audio	0.111319	RTSP live audio	0.000168
Non-P2P traces type 1	0.972249	Non-P2P traces type 1	0.280714	RTSP on-demand audio	0.021059	Telnet session	0.000000

the type 1 and type 2 traces and for the applications considered in Table 1. For the sake of simplicity, it only includes the applications for which the entropy, using a sliding window with size of 100 bytes and excluding the packets without TCP payload, was between 1 and 3. We considered the traffic with entropy greater than 3 (in the same conditions) to be P2P and the traffic with entropy lower than 1 to be non-P2P. Hence, in

this phase, the attention is focused in the *gray area* in the middle.

The results show that P2P and non-P2P traffic is almost distinguishable when the entropy is analyzed for the outgoing traffic. After the analysis of the entropy considering slots of 200 bytes, the two classes of traffic are separated almost perfectly with the exception of the web browsing traffic. The entropy level for non-P2P traffic is much lower than 0.1, whereas for P2P data entropy is always above that value.

4.3. Discussion

Looking at the traffic from a host level perspective does not have the purpose of extracting knowledge about individual flows or packets. Instead, it gives information about the behavior of individual users or nodes, which is useful for the characterization of the traffic. The study of the packet length heterogeneity may be used as a means to identify users running P2P applications and acting as both client and server. Such information would help to improve the accuracy and reduce the computation cost of flow-based classification methods. The results obtained by resorting to entropy revealed that aggregated traffic from a single host is composed by packets whose lengths are more heterogeneous when P2P applications are being used. Figure 4 and especially Table 1 show a clear distinction between P2P and non-P2P traffic. Moreover, the lengths of the packets generated by P2P VoIP applications are even more heterogeneous than the ones from P2P video streaming and P2P file-sharing.

The causes for the higher heterogeneity are distinct for P2P video streaming and P2P file-sharing and for P2P VoIP. In the cases of P2P video streaming and P2P file-sharing, each host establishes many concurrent connections with different hosts, while in P2P VoIP, the packet lengths are more varied due to the real-time nature of this kind of traffic. The results obtained for P2P VoIP traffic may depend on the speech codec used by the applications, being this issue left to be explored in a further study. Nevertheless, VoIP applications were used in this work with no concerns about the speech codec, exactly as a common user would do, so that we could emulate the normal use of these applications. According to our observations, they seem to mainly use variable bit rate codecs, resulting in the behavior explored by this study.

Based on the results obtained for sliding windows with size of 100 packets and filtering out the TCP packets without payload, it is possible to classify as P2P the cases where entropy is higher than 3, and as non-P2P the cases with entropy lower than 1. For the cases for which entropy is in-between those values, analyzing the entropy of the outgoing traffic makes it possible to differentiate the P2P traffic, as shown by the results in Table 2.

Nevertheless, there are also a few particular cases, like the traffic from web browsing, *Telnet* sessions, and IMAP email, that deserve special attention. The human behavior has a strong effect on these applications. The traffic from *Telnet* sessions is composed by packets with multiple lengths that result from each character written in the command line, as well as from the output of the commands. Usually, the larger packets are consequence of the commands output data that is sent to the host. A similar situation happens with IMAP as the traffic does not result only from email transfers. Synchronization between the local and remote accounts of operations like removing or marking messages as read, or moving messages between folders generate packets with different lengths. On the other hand, outgoing traffic is formed mostly by packets that fall into the same slot when we consider slots of 200 bytes. Hence, it is easily distinguishable from P2P traffic when we analyze the entropy of the outgoing traffic separated into slots.

In the case of web browsing, the traffic is formed by extremely brief sessions with multiple short flows. Each simple page view may be considered a session as it originates several flows that may be independent from other page views and may even be separated by several minutes. Since the flows used to download the contents of a web page have a very short duration, each of them contains only a few large packets and a last one carrying the remaining data. Although, individually, these flows have a stable and homogeneous behavior in terms of the packet lengths, their aggregation presents a high entropy due to the low percentage of large packets and the different lengths of the last packet of each flow. Therefore, in all the phases of our analysis, web traffic presented a level of entropy similar to the one obtained for P2P traffic. However, one should notice that the traces we used, contain only traffic from simple page views. Traffic from audio and video streaming, e.g, using *Flash*, was considered in other classes. Thus, the traffic volume from simple page views is very small and should not raise any concerns for traffic management. Moreover, when aggregated with the traffic from other applications, its effect in the entropy is diluted and its weight in the results decreases. Indeed, it is possible to see that, although the type 1 and type 2 traces contain traffic from web browsing, the entropy level is still differentiable from P2P traffic.

5. HOST-BASED CLASSIFICATION

In order to evaluate if the patterns described above are noticeable in real traffic, we implemented a simple host-based classifier, relying on the entropy value for sliding windows with size of 100 packets. The following heuristics were defined and implemented as illustrated by Figure 8:

1. if the entropy value, excluding the TCP packets with no payload, is greater than 3, the host is running P2P applications (E_noTCPPayload > 3);

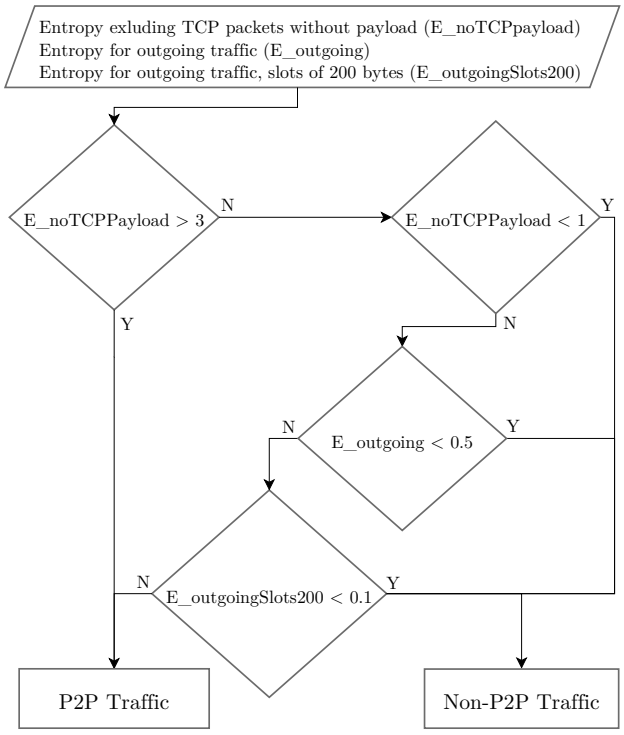

FIGURE 8. Flowchart of the proposed classification scheme.

2. if the entropy value, excluding the TCP packets with no payload, is smaller than 1, the host is not running any P2P applications (E_noTCPPayload < 1);
3. if the entropy value for outgoing traffic is smaller than 0.5, the host is not running any P2P applications (E_outgoing < 0.5);
4. if the entropy value for outgoing traffic, using slots of 200 bytes, is smaller than 0.1, the host is not running any P2P applications (E_outgoingSlots200 < 0.1);
5. in any other case, the host is running P2P applications.

The classifier processes these heuristics sequentially, meaning that each of them is used only if the previous ones were not valid.

The existence of available traces with payload is scarce. Moreover, even if we could have access to such traces, obfuscation techniques used by many applications would make it very difficult to determine which applications generated each flow. Therefore, we set up a testbed using several host computers, running different operating systems, and we captured the traffic generated by all the machines in an aggregation point. Each computer was monitored so as to be sure of the applications used in every moment and by each host. Different applications and protocols were ran in the hosts, some times simultaneously. Four datasets, with 1.8, 1.6, 3.1, and 15.5 GB, were captured and used to evaluate the effectiveness of the classification scheme proposed. All of them contain P2P and non-P2P traffic. Nevertheless, the fourth dataset contains a larger share of traffic from P2P applications. The results obtained for the classification are listed in Table 3.

The classifier based on the patterns identified in this study performed very well, with a false positives rate of almost 3% and a false negatives rate of less than 10%. Even though the analysis used in this work is completely *in the dark*, using only the information of the length of the packets, it was possible to accurately identify the hosts running P2P applications. The method used is very lightweight, it does not require the calculation of probability distributions, nor does it need to correlate the information of the packet lengths with other properties like, for example, the inter-arrival times. Moreover, since it relies only on the characteristics of the lengths of the packets generated by generic P2P applications, it can identify encrypted traffic and traffic from previously unknown P2P protocols. In fact, during the experimental tests, the classifier identified traffic from a flash-based streaming service of a TV channel (CNN) as being P2P traffic. After a human verification, we realized that what first seemed to be a false positive case was, indeed, a true positive. The CNN streaming service, which we thought to be a common *client-server* service, was using a plugin to implement a P2P system to reduce the costs of the video distribution.

6. CONCLUSION

The search for new methods that could provide a deeper knowledge about the behavior of the network traffic led the researchers to look at the traffic from different perspectives. Many approaches rely on statistical tools to describe the traffic properties mathematically and derive conclusions that can be used in practice due to their computational efficiency.

In this article, we used source traffic from individual users and based our study on a host level perspective. We analyzed the characteristics of the lengths of the IP packets from several popular applications, giving special attention to the dissimilarities between P2P and non-P2P traffic. The analysis of the datasets showed different patterns regarding the heterogeneity of the lengths, which was measured using entropy. Non-P2P traffic presented a very low entropy level when compared to the P2P datasets. In order to distinguish ambiguous cases, we also analyzed the entropy for the outgoing traffic and, to improve the results, the lengths were separated into slots. Our approach relied on a sliding window with a constant size that enables the analysis of entropy in real-time and makes it sensitive to variations in the characteristics of the traffic during the lifetime of the flows.

The heterogeneity of the packet lengths, applied at a host level, can be used for the characterization of the behavior of a user or node. The information it retrieves

TABLE 3. Results of the host-based classification.

Datasets	Traffic Volume (GB)	False Positives Rate (%)	False Negatives Rate (%)
Dataset 1	1.8	04.17	12.50
Dataset 2	1.6	00.00	09.09
Dataset 3	3.1	10.42	07.69
Dataset 4	15.5	00.00	09.88
Total	22.0	03.11	09.73

may be helpful to understand the traffic generated by a single host and its interactions with the remaining nodes. Based on the observations of the packet length heterogeneity, we defined a set of heuristics and implemented a simple classifier to identify users running P2P applications, which performed accurately. Since the classifier relies on traffic characteristics identified for generic P2P traffic rather than for a specific protocol, it can be used to identify traffic from previously unknown P2P protocols. Moreover, it can be applied to encrypted traffic as it resorts only to the lengths of the packets and does not need any encrypted data carried within the packet payload.

The analysis we describe may be extended for other perspectives. We plan to study the feasibility and gain of using the same analysis at other levels (e.g., *host-port* pairs, flows, separate the incoming and outgoing traffic, etc.) and combining the results with the ones obtained at the host level. Moreover, we intend to further develop the approach we exposed and apply it directly to the classification of individual traffic flows.

FUNDING

This work was partially supported by *Instituto de Telecomunicações*, by University of Beira Interior, and by *Fundação para a Ciência e a Tecnologia*, through the grant contract SFRH/BD/60654/2009 and the project TRAMANET: Traffic and Trust Management in Peer-to-Peer Networks with contracts PTDC/EIA/73072/2006 and FCOMP-01-0124-FEDER-007253.

ACKNOWLEDGEMENTS

The authors would like to thank David A. Carvalho for his assistance in the setup of the network testbed.

REFERENCES

[1] Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. (1994) On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, **2**, 1–15.

[2] Nychis, G., Sekar, V., Andersen, D. G., Kim, H., and Zhang, H. (2008) An empirical evaluation of entropy-based traffic anomaly detection. *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2008)*, Vouliagmeni, Greece, October, pp. 151–156. ACM, New York, NY, USA.

[3] Arlitt, M. and Williamson, C. (2007) The extensive challenges of Internet application measurement. *IEEE Netw.*, **21**, 41–46.

[4] Kind, A., Dimitropoulos, X., Denazis, S., and Claise, B. (2008) Advanced network monitoring brings life to the awareness plane. *IEEE Commun. Mag.*, **46**, 140–146.

[5] Karagiannis, T., Papagiannaki, K., and Faloutsos, M. (2005) BLINC: Multilevel traffic classification in the dark. *Proc. ACM SIGCOMM Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications*, Philadelphia, PA, USA, August, pp. 229–240. ACM, New York, NY, USA.

[6] Iliofotou, M., Kim, H., Faloutsos, M., Mitzenmacher, M., Pappu, P., and Varghese, G. (2011) Graption: A graph-based P2P traffic classification framework for the internet backbone. *Elsevier Comput. Netw.*, **55**, 1909–1920.

[7] Gomes, J. V. P., Inácio, P. R. M., Lakic, B., Freire, M. M., da Silva, H. J. A., and Monteiro, P. P. (2010) Source traffic analysis. *ACM Trans. Multimedia Comput. Commun. Appl. (ACM TOMCCAP)*, **6**, 1–23.

[8] Karagiannis, T., Faloutsos, A. B. M., and Claffy, K. (2004) Transport layer identification of P2P traffic. *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2004)*, Taormina, Sicily, Italy, October, pp. 121–134. ACM, New York, NY, USA.

[9] Tutsch, D., Babin, G., and Kropf, P. (2008) Application-layer traffic analysis of a peer-to-peer system. *IEEE Internet Comput.*, **12**, 70–77.

[10] Khakpour, A. R. and Liu, A. X. (2009) High-speed flow nature identification. *Proc. 29th IEEE Int. Conf. Distributed Computing Systems (ICDCS '09)*, Montreal, Quebec, Canada, June, pp. 510–517. IEEE Computer Society, Los Alamitos, CA, USA.

[11] Moore, A. W. and Zuev, D. (2005) Internet traffic classification using bayesian analysis techniques. *ACM SIGMETRICS Performance Evaluation Rev.*, **33**, 50–60.

[12] Erman, J., Mahanti, A., Arlitt, M., Cohen, I., and Williamson, C. (2007) Offline/realtime traffic classification using semi-supervised learning. *Elsevier Perform. Eval.*, **64**, 1194–1213.

[13] Freire, E. P., Ziviani, A., and Salles, R. M. (2008) Detecting VoIP calls hidden in web traffic. *IEEE Trans. Netw. Serv. Manag.*, **5**, 204–214.

[14] Palmieri, F. and Fiore, U. (2009) A nonlinear, recurrence-based approach to traffic classification. *Elsevier Comput. Netw.*, **53**, 761–773.

[15] Moore, A. W., Zuev, D., and Crogan, M. L. (2005) Discriminators for use in flow-based classification. Technical Report RR-05-13. Intel Research, Cambridge, UK.

[16] Bernaille, L., Teixeira, R., and Salamatian, K. (2006) Early application identification. *Proc. 2nd Conf. Future Networking Technologies (CoNEXT '06)*, Lisboa, Portugal, December, pp. 1–12. ACM, New York, NY, USA.

[17] Dainotti, A., de Donato, W., Pescapè, A., and Rossi, P. S. (2008) Classification of network traffic via packet-level hidden markov models. *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2008)*, New Orleans, LA, USA, November/December, pp. 1–5. IEEE Communications Society, New York, NY, USA.

[18] Dusi, M., Crotti, M., Gringoli, F., and Salgarelli, L. (2009) Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting. *Elsevier Comput. Netw.*, **53**, 81–97.

[19] Bonfiglio, D., Mellia, M., Meo, M., Rossi, D., and Tofanelli, P. (2007) Revealing Skype traffic: When randomness plays with you. *ACM SIGCOMM Comput. Commun. Rev.*, **37**, 37–48.

[20] Branch, P. A., Heyde, A., and Armitage, G. J. (2009) Rapid identification of Skype traffic flows. *Proc. 18th Int. Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV '09)*, Williamsburg, VA, USA, June, pp. 91–96. ACM, New York, NY, USA.

[21] Gomes, J. V. P., Inácio, P. R. M., Freire, M. M., Pereira, M., and Monteiro, P. P. (2008) Analysis of peer-to-peer traffic using a behavioural method based on entropy. *Proc. 27th IEEE Int. Performance Computing and Communications Conf. (IPCCC 2008)*, Austin, TX, USA, December, pp. 201–208. IEEE Computer Society Press, Los Alamitos, CA, USA.

[22] Li, B., Ma, M., and Jin, Z. (2011) A VoIP traffic identification scheme based on host and flow behavior analysis. *J. Netw. Syst. Manag.*, **19**, 111–129.

[23] Dhamankar, R. and King, R. (2007). Protocol identification via statistical analysis (PISA). White Paper, Tipping Point.

[24] Dorfinger, P., Panholzer, G., Trammell, B., and Pepe, T. (2010) Entropy-based traffic filtering to support real-time Skype detection. *Proc. 6th Int. Wireless Communications and Mobile Computing Conf. (IWCMC '10)*, Caen, France, June/July, pp. 747–751. ACM, New York, NY, USA.

[25] Gu, Y., McCallum, A., and Towsley, D. (2005) Detecting anomalies in network traffic using maximum entropy estimation. *Proc. 5th ACM SIGCOMM Internet Measurement Conf. (IMC 2005)*, Berkeley, CA, USA, October, pp. 345–350. USENIX Association, Berkeley, CA, USA.

[26] Wagner, A. and Plattner, B. (2005) Entropy based worm and anomaly detection in fast IP networks. *Proc 14th IEEE Int. Workshops Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE 2005)*, Linköping, Sweden, June, pp. 172–177. IEEE Computer Society, Los Alamitos, CA, USA.

[27] Han, C.-K. and Choi, H.-K. (2009) Effective discovery of attacks using entropy of packet dynamics. *IEEE Netw.*, **23**, 4–12.

[28] Androulidakis, G., Chatzigiannakis, V., and Papavassiliou, S. (2009) Network anomaly detection and classification via opportunistic sampling. *IEEE Netw.*, **23**, 6–12.

[29] Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., and Diot, C. (2003) Packet-level traffic measurements from the Sprint IP backbone. *IEEE Netw.*, **17**, 6–16.

[30] Bianco, A., Mardente, G., Mellia, M., Munafò, M., and Muscariello, L. (2009) Web user-session inference by means of clustering techniques. *IEEE/ACM Trans. Netw.*, **17**, 405–416.

[31] Shannon, C. E. (1948) A mathematical theory of communication. *The Bell System Technical J.*, **27**, 379–423.

Chapter 5

Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties

This chapter consists of the following article:

Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties

João V. Gomes, Pedro. R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

Revised version of the article submitted for publication in an international journal.

Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties

João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

Abstract—Voice over Internet Protocol (VoIP) applications based on peer-to-peer (P2P) communications have been experiencing considerable growth in terms of number of users. To overcome filtering policies or protect the privacy of their users, most of these applications implement mechanisms as protocol obfuscation or payload encryption that avoid the inspection of their traffic, making it difficult to identify its nature. The incapacity to determine the application that is responsible for a certain flow raises challenges for the effective management of the network. In this article, a new method for the identification of VoIP sessions is presented. The proposed mechanism classifies the flows, in real-time, based on the speech codec used in the session. In order to make the classification lightweight, the behavioral signatures for each analyzed codec were created using only the lengths of the packets. Unlike most previous approaches, the classifier does not use the lengths of the packets individually. Instead, it explores their level of heterogeneity in real-time, using entropy to emphasize such feature. The results of the performance evaluation show that the proposed method is able to identify VoIP sessions accurately and simultaneously recognize the used speech codec.

Index Terms—Data communications, distributed applications, network communications, network management, network monitoring, packet-switching networks.

1 Introduction

The popularity of Voice over Internet Protocol (VoIP) applications relying on the peer-to-peer (P2P) paradigm has been growing in the last few years. The simplicity of these solutions, as well as their economic benefits over the traditional telephony, make them an increasingly common choice for long distance calls and voice conferences. Furthermore, the possibility to integrate them in mobile devices, like smartphones and tablet computers, make them more flexible and easy to use. When implemented over P2P systems, VoIP applications take advantage of the scalable and reliable properties of the distributed nature of the P2P model, which puts the intelligence at the network edges.

Over the years, many of these applications have started to adopt measures to disguise their traffic and avoid the inspection of their contents. Protocol obfuscation, payload encryption, and the use of random port numbers are now common features in the majority of the popular VoIP software clients. *Skype* is the most demonstrative example of this trend: it is based on a closed code and proprietary P2P protocol, its communications are encrypted, and it has a large number of users. Nevertheless, there are also other VoIP solutions based on P2P communications that use different protocols. The Session Initiation Protocol (SIP) used by several VoIP applications or an extension of the Extensible Messaging and Presence Protocol (XMPP) used by *Google Talk* are good examples of such VoIP systems.

In most of these applications, the implementation of techniques to avoid the inspection of traffic has primarily the intention of protecting the privacy of the data of the VoIP sessions. However, it also renders difficulties for the correct and effective management of the computer networks. Understanding what kind of data is being transmitted in each flow is of critical importance to organize the network and its traffic, distribute the available bandwidth fairly, or guarantee the Quality of Service (QoS) needed by distinct classes of traffic [1], [2], [3]. Besides of the impact that VoIP applications may have in the network performance, they also raise a few security concerns. Several authors [4], [5], [6] and security institutes or companies [7], [8], [9] have exposed the potential vulnerabilities associated with VoIP systems and suggested a few guidelines to avoid security flaws.

For these reasons, traffic classification based on the application protocol has been a very active research field. The identification of VoIP, especially *Skype* related traffic, has attracted the attention of many researchers who have addressed this topic in several articles [1], [2], [3], [10], [11]. In the majority of the cases, whether the classification is made by resorting to payload inspection, flow-level heuristics, statistical analysis, or machine learning algorithms, the goal is to identify the whole data generated by the VoIP application. These flows are generated by a signaling protocol that initiates, controls, and terminates the session and by a transport protocol responsible for delivering the data from one peer to the other. The signaling data, as well as the flows used for authentication and other operations, have little impact

- *J. Gomes, P. Inácio, M. Pereira, and M. Freire are with Instituto de Telecomunicações, Department of Computer Science, University of Beira Interior, Portugal.*
  E-mail: jgomes@penhas.di.ubi.pt, {inacio, mpereira, mario}@di.ubi.pt
- *P. Monteiro is with Nokia Siemens Networks Portugal, S. A., with University of Aveiro, and with Instituto de Telecomunicações.*
  E-mail: paulo.1.monteiro@nsn.com

Manuscript received 8 May 2011.

on the network performance when compared with the data from the VoIP session. Hence, a distinct approach is followed in this work. Instead of the whole traffic from a certain VoIP application, the intention of this work is to identify the traffic from the actual VoIP session. The data transported within each packet of the session flow depends more on the speech codec used to codify the voice than on the signaling protocol or client application. In fact, the data from VoIP sessions, made using distinct applications and even distinct signaling protocols, has similar characteristics when the same codec is used. From the traffic management perspective, it may be more useful to identify the VoIP traffic with similar characteristics regardless of the application or protocol that was used, than to base the classification on the specific application that has generated it, which may include flows with different properties and purposes.

This article presents a VoIP classifier that is suitable for real-time analysis and does not rely on the payload data, being therefore applicable for encrypted traffic. Unlike most previous works, the goal of the classifier described herein is to identify the traffic flows that are related with a VoIP session in which a specific codec was used. Moreover, it our intention to minimize the number of packet-level or flow-level characteristics required to identify VoIP sessions so as to make the whole classification process lightweight. The lengths of the packets were the only traffic feature used in the identification of VoIP flows. Instead of looking at the lengths individually or calculating their mean, we focused on the relation between the different lengths and explored their level of heterogeneity using entropy. The characteristics of the packets from a VoIP session using different codecs were carefully analyzed. Several distinct applications and speech codecs were considered in the study. Based on this analysis, a set of behavioral signatures for each codec is proposed. Each of them is formed by an interval for the entropy and another one for the lengths of the packets. Additionally, a sliding window with a constant size of N packets was implemented to assess the heterogeneity in real-time and to avoid losing the sensitivity to the local changes in the values. To the best of our knowledge, the level of heterogeneity was used for the purpose of traffic classification only in our previous works [12], [13], with the exception of a recent study that has followed a similar approach [14]. Nonetheless, the heterogeneity of the lengths of the packets was only analyzed offline for complete flows.

The performance of the classification mechanism was evaluated using datasets containing traffic from VoIP sessions as well as from multiple P2P and non-P2P applications or services. The results show that the method identified the flows from VoIP sessions with very good accuracy and it was also able to recognize the speech codec with a good sensitivity rate. Moreover, the analysis of the resources used by the proposed classifier showed that its consumption grows linearly with the size of the input data.

The remainder of the paper is structured as follows. Section 2 describes the previously published related work. The analysis of speech codecs considered in the scope of this work is included in section 3. Section 4 presents the classifier and the evaluations of its performance is discussed in section 5. The last section summarizes the most important conclusions.

2 Related Work

The classification of traffic from VoIP applications has already been studied by several authors. A few studies relied on the data carried within the payload to create signatures to identify *Skype* packets [15]. In some cases, the inspection of characters in the payload is combined with statistical data, behavioral patterns, or heuristics [11], [16], [17], [18], [19]. A different approach followed by a few authors is based on the fact that the payload data from packets generated by applications that encrypt the traffic is more random. Bonfiglio et al. [10] explored the randomness of the payload data by using the *Chi Square* test and applied the method to *Skype* traffic. Additionally, they proposed a statistical classifier based on inter-arrival times and packets lengths. In [20], [21], the authors resorted to entropy to analyze the randomness level of the data from encrypted traffic.

Methods based on heuristics are also proposed in some articles [1], [22], [23], [24]. Generally, the heuristics use several flow-level or packet-level features or try to model behavioral patterns (e.g., transport protocol used in both directions, number of connections, etc). The flow or packet-level features are also analyzed statistically by some studies and used to identify VoIP traffic [2], [25].

The use of machine learning algorithms has also been applied to the traffic classification, and specifically, to the VoIP traffic identification. Jun et al. [26] proposed a method to identify *Skype* traffic based on the *Random Forest* classifier, while Branch et al. [27] relied on the *C4.5* decision tree algorithm. In [28], symbiotic bid-based genetic programming was used to identify *Skype* encrypted traffic and the performance was compared with *C4.5* and *AdaBoost* algorithms. Wu et al. [29] explored characteristics of the human behavior, as the speech period, and used a Naïve Bayes classifier to identify VoIP traffic. Zhang et al. [30] proposed a method based on Support Vector Machines (SVMs), that uses a set of traffic features to identify *Skype* communications.

The approach followed in this article resorts to the characteristics of the lengths of the packets. Several previous works have already used the lengths of the packets as one of the features employed in the traffic classification. Nonetheless, they analyzed them mostly through statistics as the mean [19] or the standard deviation [20], intervals [31], or probabilistic models [10]. On the contrary, instead of focusing on the lengths of the packets individually, the method described herein explores the relation between the different lengths by analyzing how heterogeneous these values are. Many of

the previous works that explore flow level properties, through statistical measures [24] or machine learning algorithms [30], separate the traffic into flows offline and then apply the classification approach, making these methods difficult to adapt (or even unsuitable) for the real-time analysis of the traffic. The approach followed herein implements a sliding window with size of N packets that produces information about the traffic characteristics in every step, during all the duration of the flow since its beginning.

Furthermore, besides of identifying the VoIP related data, the proposed classifier also tries to give a strong prediction on the speech codec used in a VoIP session instead of identifying the VoIP application, which is the goal of most studies. In fact, although packet or flow properties like the length of the packet or the inter-arrival time differ when distinct codecs are used, most studies seem to use them without considering the speech codec. Besides, some of the properties identified as being specific for a certain VoIP application may also apply to other applications that use similar codecs. Nonetheless, a few authors have considered the influence of distinct codecs when proposing a classification method. Branch et al. [27] analyzed the traffic from the Sinusoidal Voice Over Packet Coder (SVOPC) codec, while Molnár and Perényi [24] focused on the Internet Speech Audio Codec (iSAC). Chen et al. [25] considered iSAC and the Internet Low Bit Rate Codec (iLBC) and Yildirim et al. [31] analyzed three Constant Bit Rate (CBR) codecs, *G.711*, *G.723*, and *G.729*. Xu et al. [32] proposed a traffic classification method, based on a finite state machine, and applied it to identification of *Skype* traffic generated using SVOPC, Adaptive Multi-Rate Wideband (AMR-WB), *G.729*, and Pulse-Code Modulation (PCM). In the statistical analysis of *Skype* VoIP flows described in [33], the authors considered iSAC. A more comprehensive set of codecs, which includes iSAC, iLBC, *G.729*, Internet Pulse Code Modulation wideband (iPCMwb), Enhanced *G.711* (EG711) A/U, PCM A/U, and SVOPC, was analyzed by Bonfiglio et al. in a study of *Skype* traffic [3] and used in the classifier described in [10]. Nevertheless, none of these works presented a method to identify the codec used in a VoIP session, nor proposed signatures for each codec. Moreover, the analyzed codecs are mostly codecs used by older versions of the *Skype* software.

We proposed the analysis of the level of heterogeneity of the lengths of the packets from P2P applications and its quantification through entropy for the first time on a previous article [12]. The work described herein elaborates on that method, and evolves to the identification of VoIP traffic from different speech codecs. The most comparable work was published recently by Li et al. [14] who used a similar approach, in conjunction with an analysis of the inter-arrival times, to identify CBR and Variable Bit Rate (VBR) codecs. Their method is based on the idea that CBR codecs produce packets with constant lengths and VBR codecs produces packets with different lengths. They did not analyze the behavior of different codecs, nor try to identify the specific codec used in a session. Moreover, even the different VBR codecs may produce packets whose lengths can be more or less heterogeneous depending on the specific codec. The algorithm proposed by Li et al. is also based on the offline analysis of the traffic. The heterogeneity of the traffic is analyzed for complete flows. Besides preventing the method from being applied to real-time monitoring, their approach also raises a few problems. If the characteristics of the traffic change or occasional occurrences of different lengths appear in the middle of the flow, the results of the analysis of the heterogeneity for the whole flow may be compromised.

The work from Li et al. appears to be based on [34], in which Okabe et al. analyzed traffic from a *Skype* codec and from *G.711* and *G.723*. Instead of studying the heterogeneity of the lengths of the packets, they separated the traffic into flows, counted the number of distinct observed lengths, and used the result as a feature to identify VoIP flows offline. Liu et al. [35] explored the ratio between small packets and large packets and used that value, together with a few heuristics, to identify P2P traffic offline. Wright et al. [36] used the packet lengths for a different purpose. Instead of identifying the application or codec that generated the data, they analyzed the lengths of the packets generated by VBR codecs to try to recognize spoken phrases in encrypted VoIP sessions.

TABLE 1
Applications and codecs considered in the study.

Application	Codecs
Blink	PCM A/U, *G.722*, iLBC, GSM, *Speex*
Ekiga	PCM A/U, *G.722*, iLBC, GSM, *Speex*
Linphone	PCM A/U, GSM, *Speex*
QuteCom	PCM A/U, *G.722*, GSM, *Speex*
SIP Communicator	PCM A/U, *G.722*, GSM, *Speex*
Skype	iPCMwb, iSAC, EG711 A/U, PCM A/U, iLBC, *G.729*, AMR-WB, SVOPC, NWC, *SILK*
X-Lite	PCM A/U, iLBC, GSM, *Speex*

3 Analysis of Speech Codecs

The proposed method is based on the properties of the lengths of the packets for different codecs, regardless of the VoIP application. To understand and study the behavior of the traffic from each codec, it was necessary to collect traffic from VoIP sessions using different speech codecs. A set of applications was used to perform the calls so as to consider any possible influence of the application in the characteristics of the traffic. With the exception of *Skype*, the used applications resort to SIP for signaling. Table 1 presents a summary of the applications and codecs considered in this article. To allow the capturing of experimental data from specific codecs, we included only VoIP applications that offer the possibility of choosing the codec in a preferences menu. Moreover,

since it was our intention to use *Microsoft Windows* and *Linux* platforms in the experiences, only applications that have versions for both operating systems were selected.

3.1 Speech Codecs

The analysis presented herein and the proposed classifier are based on the speech codecs used in VoIP sessions. We studied the lengths of the packets generated by VoIP sessions using several codecs by observing several traffic samples of each of the codecs included in Table 1 and we tried to identify patterns for each of them. The speech codecs analyzed in this work are further described in section A of the supplemental material.

Speech codecs use audio processing techniques and compression algorithms to encode analog audio into digital signal. In order to turn a continuous signal into a discrete signal, codecs take samples of the analog signal. Additionally, codecs process the analog signal in frames with a limited size which contain a segment of the signal. Most codecs use different sampling frequencies (the number of samples per second) and frame sizes, which influences the amount of data transmitted in a VoIP call. For instance, *G.729* has a sampling frequency of 8 kHz with samples of 16 bits and uses a frame size of 10 ms, while PCM has a sampling frequency of 8 kHz with samples of 8 bits and the *G.711* standard does not fix a frame size. Moreover, speech codecs use different compression algorithms and some of them prioritize audio quality, while others try to minimize the bandwidth used in a VoIP session. Besides of these aspects, CBR codecs generate packets with constant lengths, while the data generated by VBR codecs depends also on the signal they are encoding. Consequently, the packets used to transmit the data created with distinct codecs through the network present different lengths.

3.2 Experimental VoIP Traffic

In order to study the properties of the packets generated by each codec, it was necessary to collect experimental traffic from each codec. By using VoIP applications that offer the possibility of choosing the codec in a preferences menu, we were able to force the use of a specific codec in each session and capture the traffic it generates. In the case of *Skype*, there are no menu options to choose the speech codec. However, *Skype* creates a `config.xml` file for each user, in which it is possible to force the use of a specific codec or to disable codecs. By using this file, we were able to obtain traffic samples of each of the codecs used by *Skype* in its different versions.

The experimental traffic used to study the properties of the data from each codec was collected from more than 160 VoIP sessions, using the selected applications. Although it was observed that one or two minutes were enough to stabilize the properties of the traffic from each codec (as mentioned in [1]), the duration of the sessions analyzed varies from 4 to 30 minutes. The collected data totals 654 MB, 274 MB from the Transmission Control Protocol (TCP) and 380 MB from the User Datagram Protocol (UDP).

The two peers were not running any application else so as to make sure that only the traffic generated by the VoIP application was captured. As most applications implement SIP, it was also possible to analyze VoIP sessions between peers using distinct applications. Usually, VoIP applications use the Real-time Transport Protocol (RTP) over UDP. *Skype* can also resort to TCP to transport the VoIP data. Hence, in some of the analyzed sessions, the UDP traffic was blocked to force the use of TCP.

The analyzed VoIP sessions were made in small and large Local Area Networks (LANs) and through commercial home links as well. The peers were running *Microsoft Windows* and *Linux* operating systems and, in a few sessions, *Skype* and *Linphone* over *Android* operating system were also used.

3.3 Expressing Heterogeneity Through Entropy

The heterogeneity of the lengths of the packets will be explored in this article. The concept of entropy introduced by Shannon in the information theory [37] will be used to assess the level of heterogeneity. Shannon presented entropy as a measure of the uncertainty of a random variate. Entropy, denoted by $H(x)$, is defined by

$$H(x) = -\sum_{i=1}^{n} p(x_i) \ln p(x_i), \tag{1}$$

where n is the number of occurrences of x, and $p(x_i)$ is the probability of the particular occurrence of x_i. For any finite number $n \in \mathbb{N}$, the maximum value $H(x)$ may attain is given by

$$H(x) = \ln n. \tag{2}$$

The value of the entropy is always a positive number. If the number of different values in the pool of samples is small, $H(x)$ is close to 0. It increases with the number of distinct occurrences under analysis. In this article, entropy is used to measure and apply the heterogeneity of the lengths of the packets from the analyzed traffic. Hereinafter, any mention of entropy refers to the entropy of the lengths of the packets.

Since, by definition, entropy is calculated for a set of values, based on the probability of each value, it is necessary to define to which set of lengths the entropy should be calculated when analyzing aggregated traffic from one or more hosts. Given the goal of identifying VoIP flows, one option would be to calculate entropy for all the packets of each complete flow. However, such approach would produce a classification only at the end of the flow, preventing its application to real-time analyses. Moreover, any characteristics resulting from occasional behaviors in the middle of a flow might compromise the results of the analysis for the complete flow. Alternatively, if the value of entropy was obtained for intervals of time, the conclusions would also depend

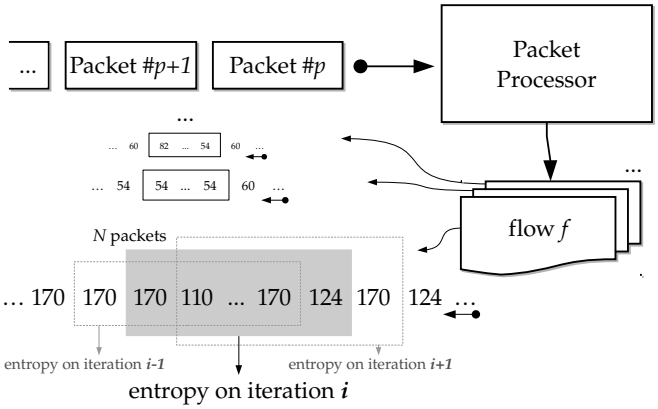

Fig. 1. An independent sliding window with size of N packets contains the lengths for each identified flow, and one entropy value is calculated in each iteration.

on the packet rate in each flow and a classification result would only be produced at the end of each interval.

Therefore, instead of calculating the entropy for each complete flow or for time intervals, it was implemented a method based on a sliding window with a constant size of N packets, as depicted in Fig. 1. For each flow, an independent window is used. Every time a new packet arrives, the flow to which the packet belongs is identified and the length of the packet is added to the corresponding window. When a new length is added, the oldest length in the window is dropped, creating the virtual movement of the sliding window.

The entropy is calculated for the lengths within the sliding window in each iteration, as exemplified in Fig. 1. One entropy value is obtained for each packet throughout the flow, with the exception of the first $N-1$ packets before the window is filled. By using this procedure, it is possible to assess the evolution of the entropy value immediately every time a new packet arrives.

In order to make the process sufficiently efficient to be used in real-time, entropy is calculated using (1) only when the window is filled for the first time. After that, instead of being calculated repetitively, its value is updated in each new iteration of the window, when the oldest length in the window (x_o) is dropped and the latest length (x_l) is added. The influence of both lengths in the entropy is updated using (3), where $p_{i-1}(x)$ and $p_i(x)$ are the probability of x in the $(i-1)$th and ith iterations of the window, respectively. The entropy value in the ith iteration of the window is calculated by updating the influence of x_o and x_l in the entropy of the previous iteration:

$$U(x) = p_{i-1}(x) \ln p_{i-1}(x) - p_i(x) \ln p_i(x), \quad (3)$$

$$H_i(x) = H_{i-1}(x) + U(x_o) + U(x_l). \quad (4)$$

3.4 Properties of the Codecs

The goal of this work is to identify the traffic from VoIP sessions and, thus, it is reasonable to focus the observation on the packets of each flow separately. The concept of flow used herein coincides with the TCP notion of connection. In the case of UDP traffic, a flow includes all the packets traveling between two *(host, port)* pairs, in both directions, with inter-arrival times inferior to 64 seconds, as suggested in [38].

However, *Skype* sometimes uses hosts, called relay nodes, that act as middle nodes mainly to overcome connection problems from users that are behind Network Address Translation (NAT) systems. We observed that, in some of these cases, it is possible to have a host receiving the incoming VoIP data from a relay node, and sending the outgoing data to a different node. The only common properties in this situation are the Internet Protocol (IP) address of the monitored host and the port used for the *Skype* session. Hence, in order to identify these VoIP connections, besides of the flow perspective, the traffic was also analyzed from the point of view of the *(host, port)* pair. This approach enables an observation level that includes all the traffic sent and received by the application process responsible for the VoIP session, even if relay nodes are used. Likewise, the analysis examples presented in this subsection concern all the traffic generated by a VoIP session, whether relay nodes are used or not.

In order to identify properties of the packet lengths from each codec, we analyzed the traffic of the VoIP sessions included in the data described in section 3.2. We observed that distinct codecs produce packets whose lengths present different levels of heterogeneity, which we measured by resorting to the entropy. As explained before, the experimental data contained sessions generated with different VoIP applications, transport protocols, and operating systems. Nevertheless, the obtained results were similar for each codec, regardless of those factors. Hence, the entropy values calculated for the different codecs were used to identify patterns.

The value of the entropy depends also on the considered window size. In [12] and, more deeply, in [13], we used sliding windows with distinct sizes from 10 to 2000 packets to analyze several datasets and we observed how the entropy varied for different sizes. The entropy may raise very slightly when the size of the window increases. Nevertheless, the most noticeable consequence of increasing the size is the stabilization of the entropy value throughout the steps of the window, which creates a smoothing effect.

This behavior is exemplified in Fig. 2. The lengths from the first three minutes of two VoIP sessions using a CBR codec and a VBR codec are represented along with the evolution of the entropy for windows with sizes of 100 and 500 packets. The y-axis ranges from 0 to the maximum value of entropy for a window of 500 packets and a dashed line is also depicted to mark the maximum value for a window with size of 100 packets. One may observe that, although for a window of 100 packets entropy is closer to its maximum, its absolute value is very similar to the entropy when using a window of 500

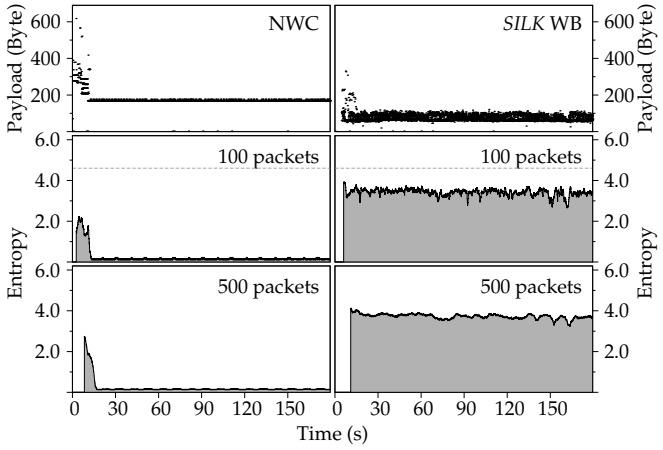

Fig. 2. Representation of the lengths of the payloads and of the entropy of the first three minutes of two VoIP sessions using NWC and *SILK* WB codecs.

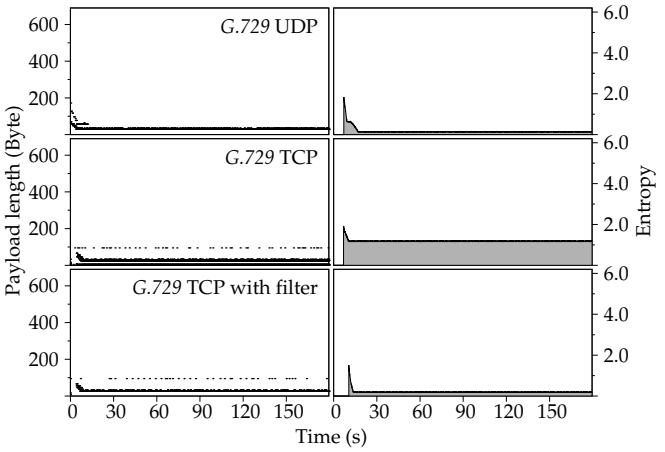

Fig. 3. Comparison of the entropy for the first three minutes of two VoIP sessions using *G.729* over UDP and TCP and the effect of filtering the packets whose transport-level payload is smaller than 5 bytes.

packets. Hence, the size of the window will not impact decisively the value of the entropy in the analyses made. Nonetheless, the stability of the entropy given by larger windows is important to identify patterns for the codecs based on the different levels of entropy. The drawback is that a larger window is likely to take more time to be filled and, consequently, it will need a few seconds more to identify a VoIP flow. Based on our previous studies [12], [13], we choose the size of 500 packets as a compromise between the stability of the entropy and the time to obtain the first result. In the appendix B of the supplemental material, we included examples of the entropy analysis using different sliding window sizes from 10 to 2000 packets.

Since the analysis performed in this work is based on the properties that result from the codec used in the VoIP session, it is useful to focus the observation only on the data carried within the transport payload. By doing so, it is possible to discard any effects of the transport protocol. Furthermore, it enables the identification of patterns that are common in the traffic from the same codec, whether UDP or TCP is used to transport the data.

Nevertheless, in some of the examples analyzed of *Skype* traffic that used TCP for VoIP sessions based on a CBR codec, it was noticed that, besides of the packets with a constant length, there were occurrences in which the length of the TCP payload was very small, being almost 0 bytes. This behavior was observable in only a few cases, and as the *Skype* protocol is closed, it is difficult to understand what is its cause. Moreover, in every case where TCP is used, there are packets whose payload has length of 0 bytes as their purpose is only to send TCP tags. These occurrences of packets with different lengths modify the heterogeneity level observed for each codec. Fig. 3 shows an example of two *Skype* sessions using *G.729*, over UDP and TCP. The lengths of the payloads from the session that resorted to TCP, are similar to the ones obtained when UDP was used. However, it is possible to observe that, when the VoIP session is made over TCP, there also several packets whose payload has length very close to 0 bytes, increasing the entropy significantly. In order to overcome this problem, the analysis mechanism implemented discharges every packet whose payload has length less or equal to 5 bytes. Using this filter makes it possible to focus on the packets that carry the voice data, obtaining a similar level of entropy to when UDP is used, as show in Fig.3.

The following subsections describe the properties identified for CBR and VBR codecs. AMR-WB is a multi-rate codec, formed by nine source codecs with distinct constant bit rates. The bit rate it uses may change every 20 milliseconds. Therefore, in spite of being a CBR codec, AMR-WB will be analyzed along with the VBR codecs. *Speex* supports CBR and VBR and, thus, examples of VoIP sessions using both modes will be analyzed with the remaining CBR and VBR codecs. The presented examples refer to analyses of the lengths of the transport-level payload, filtering out the packets whose payload is less or equal to 5 bytes and using sliding windows with size of 500 packets.

3.4.1 Constant Bit Rate Codecs

The traffic from VoIP sessions that use CBR codecs is formed mostly by packets with the same length. Hence, the entropy level is extremely low. In the case of the applications that use SIP, the entropy is almost always equal to 0 as most packets have a payload with the same length. Nevertheless, when using *Skype*, there are always a few occurrences with different lengths even if the traffic is still very homogeneous. As *Skype* uses its own closed protocol, it is difficult to understand why this happens.

In Fig. 4, one can observe a comparison between the first three minutes of VoIP sessions that used PCMA, PCMU, and iLBC, through *Skype* and SIP clients. Although in both sessions the payloads have the same

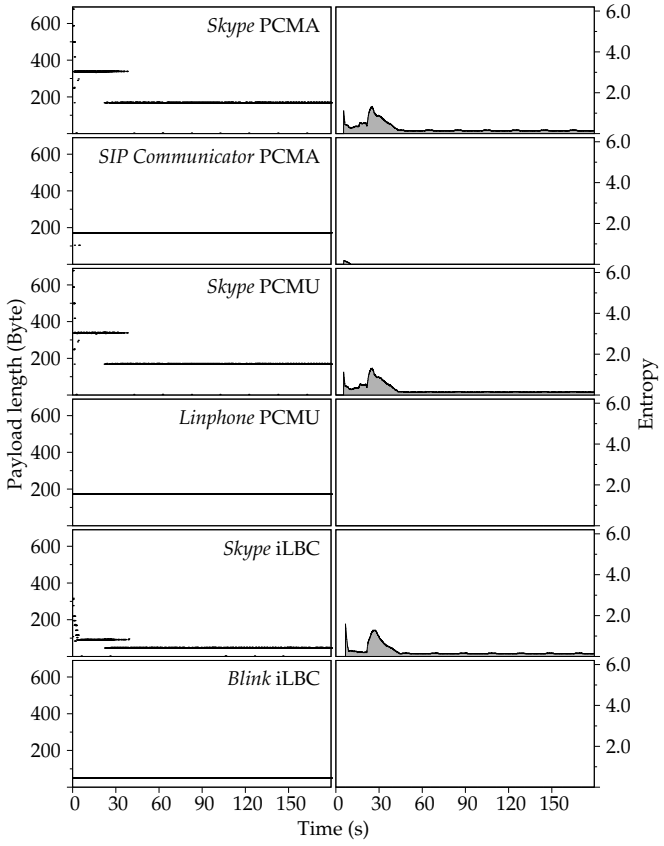

Fig. 4. Comparison of the lengths of the payloads and of the entropy between VoIP sessions using *Skype* and SIP applications with CBR codecs.

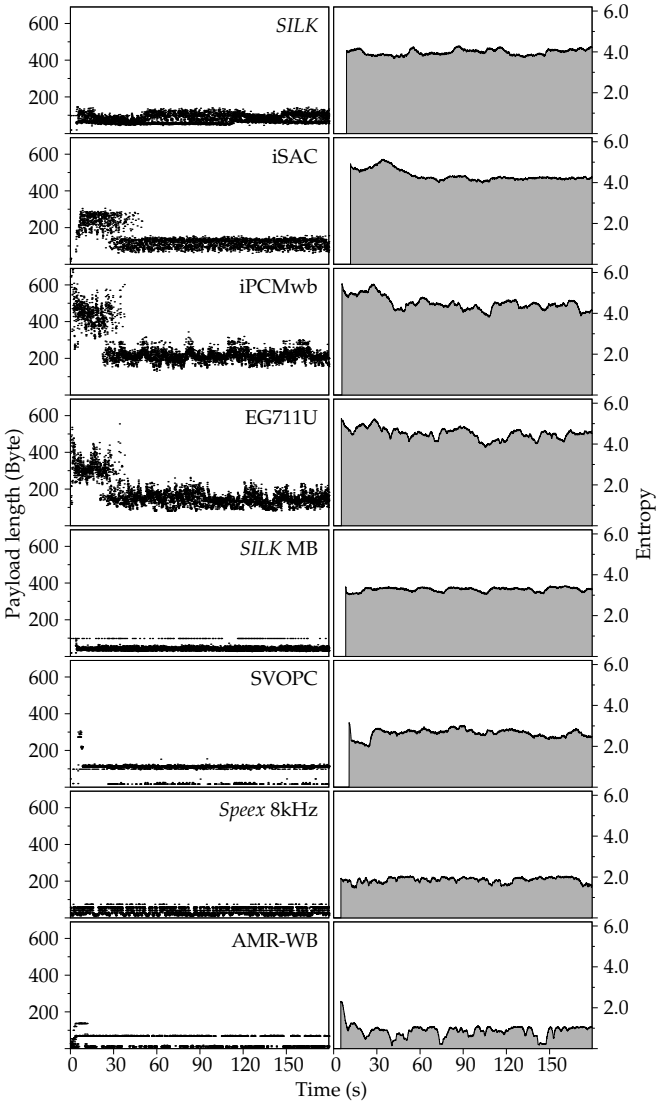

Fig. 5. Representation of the lengths of the payloads and of the entropy of the first three minutes of VoIP sessions using different VBR codecs.

length, in the case of *Skype* there are also a few packets whose payload has a different length. This behavior was observed in all the analyzed VoIP sessions in which CBR codecs were used.

Due to the limitations of space in the main article and to the large number of charts that would be needed to represent every session example that we analyzed, we included Table 1 in the supplemental material to give a general view of the obtained values for the datasets described in section 3.2. It contains the mean of the entropy for all the VoIP sessions that used each CBR codec, as well as the most frequent lengths of the transport-level payload that were observed. For each VoIP session, the mean of the entropy in all the steps of the window was calculated, which results in one entropy value for session. Afterwards, the mean of the values obtained for all the sessions in which the same codec was used was calculated and included in the table.

Entropy was analyzed separately for the incoming and the outgoing data. VoIP flows usually have similar properties in both directions, which is also important to distinguish it from the traffic of other applications. The summary included in Table 1 of the supplemental material show this similarity between the traffic in both directions. The difference between the values for *Skype* and SIP applications, described in the previous section, is also visible in the table.

The payloads of the packets from codecs based on PCM have similar lengths. Although it was not possible to find any information regarding NWC, the packets from VoIP sessions based on it also have lengths similar to the ones based on PCM.

3.4.2 Variable Bit Rate Codecs

Unlike the CBR codecs, the traffic from each VoIP session in which a VBR codec is used is formed by packets whose payloads have very heterogeneous lengths. Fig. 5 depicts the lengths of the transport-level payloads and the corresponding entropy of the first three minutes of several VoIP sessions, each of them using a different VBR codec. In all the cases, the variety of distinct lengths form a strip of values. The different levels of heterogeneity

of the payload lengths are demonstrated by the distinct levels of entropy depicted in the charts.

There also other details that are visible in the charts. In the beginning of the VoIP sessions in which the Global IP Solutions (GIPS) VBR codecs (EG711 A/U, iSAC, and iPCMwb) were used, the lengths of the payloads appear on a strip of higher values. Before the 30 seconds, they stabilize on lower values. This behavior was also observed in other VoIP sessions in which *Skype* was used, even with CBR codecs as shown in Fig. 4. In the case of *Speex*, the lengths vary within a small range of values, resulting in a lower level of entropy. This is even more evident in the case of AMR-WB, in which the entropy is even lower and less stable.

A summary of the results obtained for all the VoIP sessions from the datasets described in section 3.2 in which VBR codecs were used is presented in Table 2 of the supplemental material. The values were obtained in the same way as it was done for the CBR codecs. In the case of the VBR codecs, the table includes ranges of frequent length instead of individual values as the lengths are heterogeneous. The GIPS VBR codecs generate lengths with higher entropy. *SILK* and *SILK* WB seem to have similar properties, as well as *SILK* mediumband (MB) and *SILK* narrowband (NB), and *Speex* 32 kHz and 16 kHz. AMR-WB presents a very low entropy when compared with the VBR codecs, which was expectable since it is not a truly VBR codec.

4 The VoIP Classifier

The classifier proposed herein is based on the properties described in the previous section. The following subsections provide a list of the proposed signatures and describe the classification mechanism and its operation.

4.1 Behavioral Signatures for the Codecs

A set of behavioral signatures was defined to model the properties described in section 3.4, which result from the observation of the datasets described in section 3.2. The signatures are formed by the codec description, an interval in which the entropy should be contained, an interval in which the payload length should be contained, and a minimum number of occurrences matching these conditions so that a tuple can be classified as a VoIP session.

Table 2 lists the signatures proposed in this work and used by the classifier to identify VoIP sessions. The values defined for the intervals and for the minimum matches constant were optimized for sliding windows with size of 500 packets. Three different levels of signatures were defined. Most of them are signatures created to identify specific speech codecs. Nevertheless, signatures to simply identify VoIP sessions based on CBR, VBR codecs, and VBR codecs with low variation, or other groups of codecs, were also created. Separating the classification into a smaller number of categories improves its accuracy and makes the process faster.

TABLE 2
List of the behavioral signatures, for sliding windows with size of 500 packets, used to identify the VoIP sessions.

Signature Description	Intervals Lengths		Intervals Entropy		Minimum Matches
Bit rate level					
CBR	15	400	0.00	1.00	400
VBR (low variation)	10	400	1.25	3.25	450
VBR	15	800	2.80	6.00	450
Group level					
GIPS VBR	75	700	3.50	5.50	400
PCM based	160	190	0.00	1.00	400
Skype CBR	25	190	0.10	1.00	400
Skype proprietary VBR	20	120	1.50	4.50	400
Codec level					
G.722	171	175	0.00	0.10	450
G.729	25	30	0.10	0.95	400
GSM	44	45	0.00	0.10	450
iLBC	49	51	0.00	0.10	450
iLBC	87	90	0.00	0.10	450
iLBC *Skype*	46	50	0.05	1.00	400
iLBC *Skype*	86	90	0.05	1.00	400
NWC	160	171	0.10	1.00	450
PCM	165	185	0.00	0.10	450
PCMA *Skype*	160	171	0.10	1.00	450
PCMU *Skype*	170	185	0.10	1.00	450
Speex 32 kHz	85	87	0.00	0.10	450
Speex 32 kHz	45	50	0.00	0.10	450
Speex 16 kHz	80	85	0.00	0.10	450
Speex 16 kHz	40	45	0.00	0.10	450
Speex 8 kHz	50	52	0.00	0.10	450
Speex 8 kHz	30	35	0.00	0.10	450
AMR-WB	45	80	0.15	1.75	250
EG711	200	550	3.00	5.50	400
EG711	75	250	3.50	5.50	400
iPCMwb	250	700	3.00	5.50	400
iPCMwb	150	300	3.50	5.50	400
iSAC	100	300	3.00	5.50	400
iSAC	60	200	3.50	5.50	400
SILK	40	120	2.75	4.50	250
SILK MB/NB	20	60	2.00	3.50	400
SVOPC	80	120	1.50	3.00	400
Speex	20	100	2.00	2.50	400
Speex	20	100	1.50	2.00	400

4.2 Architecture of the Classifier

The implementation of the classifier includes two alternative levels of observation, as explained in section 3.4: flow or *(host, port)*. In order to individually identify each flow or *(host, port)* pair, the classifier uses an identification tuple. When the flow perspective is used, the tuple is formed by the source IP address and port number, by the destination IP address and port number, and by the transport protocol (UDP or TCP), whereas, for the *(host, port)* pair perspective, the tuple is formed by the host IP address and the port number. In this section, we will use the term *tuple* to designate a generic flow or *(host, port)*

pair, depending on which perspective is used.

In the analysis of VoIP sessions described in section 3, we observed that the heterogeneity of the packet lengths is similar in both directions. On the contrary, for other types of application that may use speech codecs, like audio streaming, the packet lengths may present the heterogeneity associated with a speech codec only in one direction, while the traffic in the opposite direction is mainly formed by acknowledgement messages. Hence, to avoid these cases, the classifier separately analyzes the VoIP session traffic in each direction and only if the packet lengths in both directions have similar properties, the session is classified.

Furthermore, as described in section 3.4, the traffic analysis showed more than one frequent length for some codecs, which results in more than one signature for the same codec in Table 2 (e.g., iLBC codec). We observed that the traffic from some VoIP sessions that use one of those codecs has distinct packet lengths in both directions. For example, in some sessions using the iLBC codec, the packets in one direction had 46 bytes, while in the opposite direction, the packets had 86 bytes. Hence, one of the two iLBC signatures included in Table 2 matches one direction of the VoIP session traffic, while the other signature matches the traffic in the opposite direction. Therefore, when separately analyzing the traffic in each direction, the classifier tries to classify the traffic in both directions with signatures for the same codec, even if the signatures are distinct signatures for the same codec.

The proposed classifier is formed by three modules: one responsible for processing the packets, other for calculating the entropy level, and a third one for identifying the VoIP data. The architecture and operation of the mechanism and the process of identification of a VoIP session are explained in the following subsections.

4.2.1 The Modular Operation

The modular architecture of the mechanism is illustrated by Fig. 6. The packets are received, from a live or an offline source, by the *packet processor* module, which extracts the transport-level payload length and the tuple identifier based on the perspective used in each analysis of the classifier. Additionally, the *packet processor* filters out every packet whose transport layer payload is smaller than or equal to 5 bytes, as explained in section 3.4.

A statistical analysis module was implemented to calculate several statistics based on the sliding window method depicted in Fig. 1 and it was used to perform the VoIP sessions analysis described in section 3. The same module was also used as one of the components of the classifier. It receives the tuple identification and the length of the payload from the *packet processor* module, includes it in the corresponding sliding window, updates the statistics, and returns the value of the entropy, in the latest step of the window for the considered tuple, to the *packet processor*. This module receives the data from

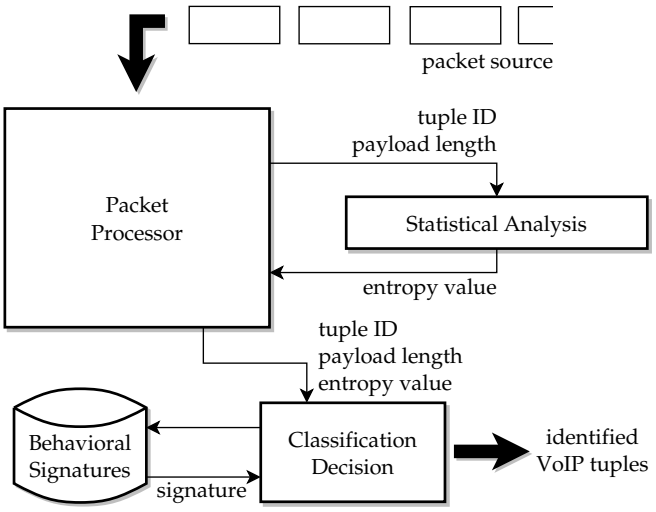

Fig. 6. Architecture of the proposed classifier formed by three modules.

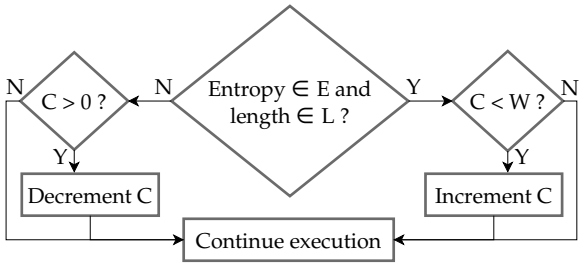

Fig. 7. Signature matching process used by the *classification decision* module.

the *statistical analysis* module and, if the sliding window of the corresponding tuple is already filled, it sends the identification of the tuple, the length of the payload, and the entropy value to the *classification decision* module.

4.2.2 The Classification Decision Module

The *classification decision* module receives, from the *packet processor*, the payload length and the entropy value along with the identification of the corresponding tuple and produces a classification result. The classification process is formed by two main tasks, the signature matching and the classification based on the matched signatures, and is repeated for every processed packet.

During the signature matching process, the module tries to match all the behavioral signatures in the repository. As explained in section 4.1, each signature S, associated with a codec Cod, is formed by two intervals E and L to which the entropy and the packet length should belong, respectively, and by the required minimum number of matches $minM$ in the latest W (size of the sliding window) packets so that the tuple can be classified as traffic generated by Cod. For each pair formed by S and the analyzed tuple T, there is an individually counter C of the number of matches in the last W, whose value is always between 0 and W. The classifier tries to match

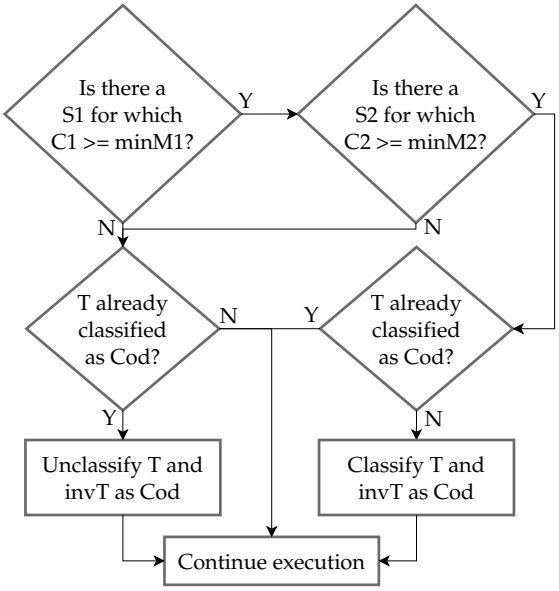

Fig. 8. Classification process based on the results of the signature matching.

each signature *S* in the repository, as depicted in Fig. 7. Depending on the result of the signature matching, the *C* counter associated with *S* and *T* is decremented or incremented, unless it is already 0 or *W*, respectively.

After all the signatures are checked, each counter *C* associated with each signature *S* contains the number of matches of *S* for tuple *T*, allowing the classifier to make a decision using the method represented in Fig. 8. The classifier goes through the signatures repository and checks if, for each codec *Cod*, there is a signature *S1* with required minimum number of matches *minM1* so that the corresponding counter *C1* is greater than *minM1*. If it does, the classifier has to check if the signature for the same codec also matches the traffic in the opposite direction, identified by the inverse tuple *invT*. Since there are different signatures for the same codec, as explained in the beginning of section 4.2, the classifier has to check if there is a signature *S2* (which can be the same as *S1*) for the same codec *Cod* with a minimum number of matches *minM2* so that the corresponding counter *C2* for *invT* is greater than *minM2*. In case both conditions are true, *T* and *invT* are classified as traffic from codec *Cod*, if they have not been before. Otherwise, *T* and *invT* are *unclassified* as traffic from *Cod* if they have been classified before, meaning that they do not present characteristics of *Cod* anymore, probably because the VoIP session has finished.

5 Performance Evaluation

The evaluation of the classifier was made by resorting to offline data so that the procedure could be repeated and compared against other classifiers. The following subsections describe the datasets used in the evaluation of the classifier and the obtained results.

5.1 Datasets

Evaluating the performance of a traffic classifier is not an easy task as it is necessary to previously know which application has generated each flow in the used traffic samples. Since there are not many available datasets labeled with the ground-truth information, some of the studies proposing new traffic classifiers use a Deep Packet Inspection (DPI) mechanism as a reference classifier. Such approach sometimes goes against the motivation of the studies that claim that new classifiers are necessary since DPI is becoming ineffective due to encryption and other evasive techniques. Besides, available datasets do not usually contain payload data which renders DPI useless as a reference classifier.

Moreover, the approach followed herein is focused on the codecs used in each VoIP session. In order to evaluate the accuracy of the codec prediction, it would be necessary to have datasets containing VoIP sessions and labeled with the information of the codec used in each of the sessions. To the best of our knowledge, the only datasets of VoIP sessions separated by the speech codec were made available by the Telecommunication Networks Group of the Politecnico di Torino, in the website of *Tstat* [39]. Although these traces are a good resource, they only contain a small number of VoIP sessions of a subset of the codecs used by *Skype*, which is insufficient to evaluate the performance of the classifier. To overcome this problem, a testbed was setup to collect traffic from VoIP sessions and keep a record of the codecs and applications that were used. Using this approach, we collected four datasets containing 1.7, 1.6, 3.1, and 15.5 GB of data, as described in Table 3 of the supplemental material. In order to maximize the possibility of having false positive cases, other common classes of applications were used at the same time during the datasets capturing. More details regarding the testbed and the datasets are provided in appendix D.1 of the supplemental material. The traces from Politecnico di Torino (Polito) contain only the flows used by the VoIP sessions and were also used in the analysis.

5.2 Accuracy of the Classification

The collected datasets were processed by the proposed classifier. In order to evaluate its accuracy, the results were compared with the ground-truth information gathered at the moment of the capture. For each dataset, the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) cases (flows) were counted. Based on these values, two metrics were used to measure the accuracy of the classifier, *sensitivity* and *specificity*, defined by (5) and (6), respectively, as described in [40] and [41]. Sensitivity evaluates the ability of the method to identify the existent VoIP flows, while specificity measures its capacity to avoid false positive classifications.

$$Sensitivity = \frac{TP}{TP + FN} \quad , \tag{5}$$

$$Specificity = \frac{TN}{TN + FP} \quad . \tag{6}$$

The accuracy of the classifier was evaluated separately for the behavioral signatures of the bit rate, group, and codec levels, and the results are listed in Table 3. The proposed method continually analyzes every packet since the beginning of the flow and it makes a classification as soon as the properties of the packet lengths are matched by one of the signatures. Nevertheless, especially in the case of *Skype* traffic, those properties are not always very stable in the beginning of the connection. Hence, although the traffic is initially matched by a signature, seconds later the properties of the packet lengths are more stable and slightly different and are thus matched by a different signature. Since the classifier continues to analyze every packet, it modifies the classification when the traffic is matched by a different signature. This usually happens for similar signatures, such as *SILK* and *SILK* MB/NB or VBR and VBR (low variation), and it is observable, e.g., in dataset 2. For this reason, the evaluation of the sensitivity for the second classification was also included in Table 3. Nonetheless, if for a certain flow, the classifier cannot establish a stable classification, we considered it a false negative case. The average time used by the classifier to reach the first classification result, and in some cases the second, is presented in Table 6 of the supplemental material.

Generally, the sensitivity decreases from the bit rate level to the codec level as the signatures are less broad on the latter. For the same motive, the specificity decreases in the opposite direction. Nevertheless, there are a few exceptions. In the case of the bit rate level signatures, the traffic from low variation VBR codecs was, for a few sessions, classified initially as VBR and only a few seconds later as VBR (low variation). Also, the traffic from VBR *Speex*, which is not covered by any group level signature, was sometimes classified by the signature for *Skype* proprietary VBR, especially in dataset 3, which pulled down the sensitivity rate. Since, the dataset of the Politecnico di Torino contains only the flows that result from the VoIP sessions, it does not have any negative case. Hence, it does not make sense to calculate the specificity for this dataset. The percentage of the traffic in datasets 1, 2, 3, and 4, from each class of applications, that caused false positive cases is presented in Table 7 of the supplemental material, showing that they were caused by streaming, P2P file-sharing, and P2P streaming traffic.

The results show that the method is capable of classifying the traffic from VoIP sessions and identifying the used speech codec with interesting accuracy. Moreover, we obtained similar results for the same speech codec despite the fact that the datasets used for the performance evaluation contained traffic from VoIP sessions generated with different applications, transport protocols, and operating systems, showing the independence of the classifier from these factors. The performance of the classifier proposed herein was also compared with the results obtained with other available classifiers. Although there are a few studies proposing methods for the identification of VoIP traffic, there are not many implementations available. A platform to compare the performance of traffic classifiers, named *NeTraMark*, has been recently released [42]. *NeTraMark* incorporates a few classifiers, based on different methods, that classify the traffic into several classes. However, none of them is prepared to identify VoIP traffic. Furthermore, there is no classifier capable of identifying, or at least making a strong prediction, of the codec used in a VoIP session. Hence, it is not possible to make a direct comparison with the method described in this article.

Even so, three different available tools were chosen for distinct reasons and tested with the same datasets. *l7-filter* [43] was used as an example of a DPI classifier based on payload string signatures. *l7-netpdlclassifier* [44], another DPI tool, classifies the traffic based on the structure of the packets. It resorts to a list of protocol descriptors defined using *NetPDL* [45], a language developed for packet header description and extended for traffic classification purposes. *Tstat* [39] is mainly a tool for the statistical analysis of traffic. Nonetheless, it incorporates the behavioral method proposed in [10], which is based on Naïve Bayesian classifiers and does not resort to the data carried in the payloads. *Tstat* was tested with the traffic models that come with the source code.

Table 4 contains the sensitivity and specificity results obtained for the three classifiers. The results were obtained in a similar way to the ones of the proposed classifier. We counted the true and false positive and negative cases by identifying the VoIP and the non-VoIP flows that were corrected classified and the ones that were misclassified. Unlike the approach followed by this article, these classifiers are not always focused on the specific flow generated by each VoIP session. In many cases, the classifiers were not able to classify the session flow, but they correctly identified signaling flows generated by *Skype* or SIP protocols. Hence, the results included in Table 4 were calculated by considering as *true positives* two distinct cases: only the VoIP session flows identified; or also the signaling flows correctly classified as VoIP. Moreover, *Tstat* distinguishes the *Skype* traffic between two computers, the traffic between *Skype* and traditional telephony, and the *Skype* signaling data. Sometimes during the evaluation, although *Tstat* was able to identify the specific flow that carries the conversion, it classified it as signaling. These cases were also considered separately in Table 4.

The results demonstrate that most classifiers have difficulties to identify the specific flows related with conversations, even when they are able to identify other flows of the VoIP application like signaling data. The three mechanisms seem to have more problems to identify the traffic from VoIP sessions over TCP, as shown by the low sensitivity rates for the dataset 4. The specificity rates for the same dataset are also lower, mostly due to the larger share of traffic from other P2P applications. *Tstat*

TABLE 3
Results of the performance evaluation of the VoIP classifier for the different levels of signatures.

Dataset	Bit rate level			Group level			Codec level		
	Sensitivity		Specificity	Sensitivity		Specificity	Sensitivity		Specificity
	first	second		first	second		first	second	
Dataset 1	92.31%	100.00%	99.97%	100.00%	100.00%	100.00%	84.62%	92.31%	100.00%
Dataset 2	92.86%	100.00%	99.99%	100.00%	100.00%	100.00%	78.57%	100.00%	100.00%
Dataset 3	100.00%	100.00%	99.98%	80.00%	80.00%	100.00%	93.34%	93.34%	100.00%
Dataset 4	96.97%	96.97%	99.51%	96.97%	96.97%	99.56%	84.85%	84.85%	99.99%
Polito	100.00%	100.00%	not applicable	100.00%	100.00%	not applicable	70.00%	100.00%	not applicable

TABLE 4
Results of the performance evaluation of other available classifiers.

Dataset	*l7-filter*			*l7-netpdlclassifier*			*Tstat*		
	Sensitivity		Specificity	Sensitivity		Specificity	Sensitivity		Specificity
	session	signaling		session	signaling		session	signaling	
Dataset 1	30.77%	30.77%	96.50%	69.23%	76.92%	98.04%	00.00%	76.92%	99.89%
Dataset 2	00.00%	100.00%	93.98%	100.00%	100.00%	98.11%	07.14%	57.14%	99.91%
Dataset 3	00.00%	100.00%	95.39%	100.00%	100.00%	97.37%	00.00%	100.00%	99.88%
Dataset 4	03.03%	54.55%	78.11%	09.09%	45.45%	97.73%	00.00%	12.12%	98.82%
Polito	00.00%	00.00%	not applicable	90.00%	90.00%	not applicable	00.00%	70.00%	not applicable

seems to be more conservative in the identification of VoIP traffic, which also helps it to perform better in terms of false positive cases. The Polito dataset contains only 10 VoIP sessions, and therefore any misclassified flow has an immediate negative impact in the sensitivity. Since the packets in this dataset do not contain the payload data, *l7-filter* was unable to identify any VoIP flow.

Generally, the classifier proposed herein presents a better accuracy when it distinguish between CBR, VBR, and VBR with low variation, and it is also able to make a prediction with good accuracy of the codec used in each session. Furthermore, the accuracy of the identification of the flow of the real conversation is much higher than for the other classifiers.

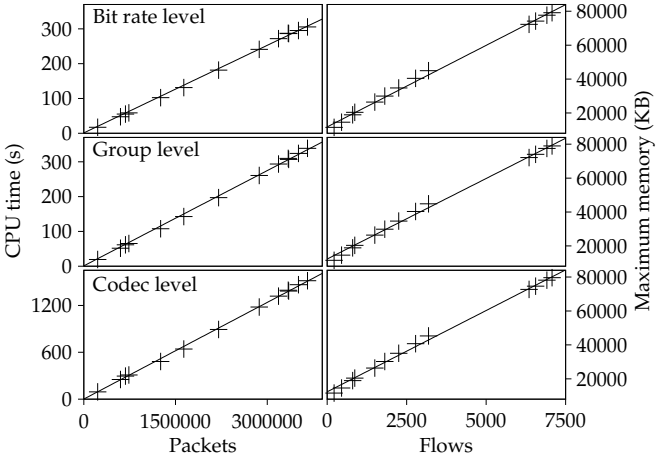

Fig. 9. Representation of the CPU time and memory consumption growing and the number of packets and flows for 13 trace files, considering packets with payload larger than 5 bytes.

5.3 Computational Resources

The computational requirements of the other classifiers analyzed in the scope of this work have, by construction, a linear dependence on the number of signatures and on the number of packets. For the sake of completeness, a simple exercise regarding the processing and memory requirements of the method described herein was made and discussed in this section. The included empirical results concern worst case scenarios. For example, the memory requirements were taken for the maximum memory the application needed during the execution.

The proposed classifier analyzes every packet that arrives to the capture point, with the exception of the packets whose transport layer payload is smaller than or equal to 5 bytes, and tries to produce a classification for each flow. The incoming traffic is separated by flows, and only a fixed number of packets is stored for each flow (fixed sized analysis window). The assignment of a packet to a flow is a negligible operation in terms of processing, and the calculation of the entropy comprises a fixed number of instructions. When a packet arrives, it is inserted in the computational representation of the respective flow, and the program tries to find the signature to which the packet belongs. In other words, each time a packet arrives, the execution goes through all the signatures. As such, the processing requirements of the classifier are linearly proportional to n_s and to n_p, where n_s is the number of signatures in the list and n_p is the number of processed packets with transport layer payload larger than 5 bytes.

The classifier only needs to save information that is

related with the flows that are active at each instant. Besides of the window of values that is stored for each flow, it also keeps a counter of the number of positives matches for each *flow, signature* pair. Hence, the memory requirements of the classifier depends only on the number of active flows in a given time instant (n_f), on the number of signatures (n_s), and on the size of the sliding windows (w). Since the *packet processor* filters out every packet whose transport layer payload is smaller than or equal to 5 bytes, n_f includes only the number of flows than contain packets with payload larger than 5 bytes. The dependence between the amount of memory and n_f, n_s, and w is linear.

The implementation of the described mechanism is far from being optimized. Nonetheless, simple measurements were made so as to have a perception of how the use of the resources by the proposed classifier grows for input data with different sizes. We performed a few experiments using `/usr/bin/time` (a tool that summarizes the system resources used by a program) for 13 distinct trace files, extracted from the four datasets used in the performance evaluation, containing a different number of flows and packets. The results of the CPU time and of the maximum memory used during the execution of the classifier for each of the trace files, using the three levels of signatures, are described in Table 8 of the supplemental material. The linear dependency between the CPU time and n_p and n_s is visible in Fig. 9.

In order to improve its memory efficiency, the classifier removes the data related to a flow when a connection ends or reaches a timeout limit. For this reason, and since the number of active flows varies throughout the trace as the connections start and finish, it would be difficult to make any useful observation regarding the used memory. Therefore, for the purpose of this exercise, we deactivated the option to remove the data from inactive flows. Using this approach, we are considering the worst case scenario in which all flows in the trace files are active flows. The maximum memory used by the proposed classifier depends only on n_f, n_s, and w. Nevertheless, it is possible to observe in Fig. 9 that the traces with less flows are using slightly less memory, which results from the fact that, as explained in section 4.2.1, the *classification decision* module only processes flows after the sliding window is filled, avoiding the need to save information for short flows with less packets than the size of the window. Since the CPU time and the memory consumption grow linearly, the mechanism may be applied for the real-time analysis of the traffic in computer networks.

6 Conclusion

In this article, a new method for the identification of P2P VoIP traffic was described. Unlike most approaches, the proposed mechanism is focused on the properties of the speech codec used in the VoIP session instead of the application and it aims to identify the flow used for the conversation rather than the signaling data. The traffic from several VoIP sessions, using many codecs and made using different applications was collected and analyzed to identify properties that could be used in the classification process. The lengths of the payloads presented different levels of heterogeneity for distinct codecs. Although the lengths of the packets have already been used in different ways, its level of heterogeneity has never been used for the classification of traffic in real-time. To the best of our knowledge, this is the first behavioral method capable of identifying the codecs used on a VoIP session.

In order to quantify the level of heterogeneity and use it to identify traffic, an approach based on entropy was used. Its value was calculated by resorting to sliding windows with size of a constant number of packets. By doing so, it is possible to monitor the value of the entropy, in real-time, from the beginning of the flow to its end. The identification of VoIP sessions is made by using a set of behavioral signatures that are formed by an interval for the entropy and an interval for the length of the payload. For each packet arriving to the classifier, a signature is matched if both the length of the payload and the entropy value are contained in the corresponding intervals. Moreover, each signature also contains a minimum number of matches that should be reached for a tuple to be classified by the signature. The list of signatures may also be extended or adapted to cover more codecs.

The performance of the classifier was evaluated, based on the proposed signatures, by resorting to aggregated traffic from multiple VoIP sessions, using different codecs and applications, and several P2P and non-P2P applications. The results showed that the classifier was capable of identifying the VoIP sessions with very good accuracy, performing better that the remaining analyzed tools. Furthermore, the mechanism was able to recognize the specific speech codec that was used with a sensitivity rate between 70.00% and 93.34%. A simple evaluation of computational resources used by the classifications showed that the consumption of resources grows linearly with the analyzed data, making the mechanism suitable for real-time analysis.

Acknowledgments

We would like to thank David A. Carvalho for his assistance in the setup of the network testbed. This work was partially supported by University of Beira Interior, by *Instituto de Telecomunicações*, and by the portuguese *Fundação para a Ciência e a Tecnologia*, through the grant contract SFRH/BD/60654/2009 and the project TRAMANET: Traffic and Trust Management in Peer-to-Peer Networks with contracts PTDC/EIA/73072/2006 and FCOMP-01-0124-FEDER-007253.

REFERENCES

[1] K. Suh, D. R. Figueiredo, J. Kurose, and D. Towsley, "Characterizing and detecting Skype-relayed traffic," in *Proc. 25th IEEE Int. Conf. Computer Communications (INFOCOM 2006)*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[2] E. P. Freire, A. Ziviani, and R. M. Salles, "Detecting VoIP calls hidden in web traffic," *IEEE Trans. Netw. Service Manag.*, vol. 5, no. 4, pp. 204–214, Dec. 2008.

[3] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi, "Detailed analysis of Skype traffic," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 117–127, Jan. 2009.

[4] J. F. Ransome and J. W. Rittinghouse, *Voice over Internet Protocol (VoIP) Security*. Digital Press, Nov. 2004, ch. VoIP Security Risks, pp. 181–233.

[5] J. Seedorf, "Security challenges for peer-to-peer SIP," *IEEE Netw.*, vol. 20, no. 5, pp. 38–45, Sep./Oct. 2006.

[6] R. Dantu, S. Fahmy, H. Schulzrinne, and J. Cangussu, "Issues and challenges in securing VoIP," *Elsevier Comput. Security*, vol. 28, no. 8, pp. 743–753, Nov. 2009.

[7] D. R. Kuhn, T. J. Walsh, and S. Fries, "Security considerations for voice over IP systems," National Institute of Standards and Technology, Gaithersburg, MA, USA, Tech. Rep. 800-58, Jan. 2005.

[8] T. Berson, "Skype security evaluation," Anagram Laboratories, Tech. Rep. ALR-2005-031, October 2005.

[9] J. Xin, "Security issues and countermeasure for VoIP," *White Paper, SANS Institute, Information Security Reading Room*, 2007.

[10] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: When randomness plays with you," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Oct. 2007.

[11] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Skype-Hunter: A real-time system for the detection and classification of Skype traffic," *Int. J. Commun. Syst.*, 2011.

[12] J. V. P. Gomes, P. R. M. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "Analysis of peer-to-peer traffic using a behavioural method based on entropy," in *Proc. 27th IEEE Int. Performance Computing and Communications Conf. (IPCCC 2008)*, Austin, TX, USA, Dec. 2008, pp. 201–208.

[13] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Exploring behavioral patterns through entropy in multimedia peer-to-peer traffic," *The Computer Journal*, accepted for publication.

[14] B. Li, M. Ma, and Z. Jin, "A VoIP traffic identification scheme based on host and flow behavior analysis," *J. Netw. Syst. Manag.*, vol. 19, no. 1, pp. 111–129, Mar. 2011.

[15] Y. Yu, D. Liu, J. Li, and C. Shen, "Traffic identification and overlay measurement of Skype," in *Proc. Int. Conf. Computational Intelligence and Security*, Guangzhou, China, Nov. 2006, pp. 1043–1048.

[16] S. Ehlert and S. Petgang, "Analysis and signature of Skype VoIP session traffic," Fraunhofer FOKUS, Berlin, Germany, Tech. Rep. NGNI-SKYPE-06b, Jul. 2006.

[17] P. Svoboda, E. Hyytiä, F. Ricciato, M. Rupp, and M. Karner, "Detection and tracking of Skype by exploiting cross layer information in a live 3G network," in *Proc. 1st Int. Workshop Traffic Monitoring and Analysis (TMA '09)*, ser. LNCS, vol. 5537, Aachen, Germany, May 2009, pp. 93–100.

[18] F. Lu, X.-L. Liu, and Z.-N. Ma, "Research on the characteristics and blocking realization of Skype protocol," in *Proc. Int. Conf. Electrical and Control Engineering (ICECE 2010)*, Wuhan, China, Jun. 2010, pp. 2964–2967.

[19] D. Zhang, C. Zheng, H. Zhang, and H. Yu, "Identification and analysis of Skype peer-to-peer traffic," in *Proc. 5th Int. Conf. Internet and Web Applications and Services (ICIW 2010)*, Barcelona, Spain, May 2010, pp. 200–206.

[20] R. Dhamankar and R. King, "Protocol identification via statistical analysis (PISA)," *White Paper, Tipping Point*, 2007.

[21] P. Dorfinger, G. Panholzer, B. Trammell, and T. Pepe, "Entropy-based traffic filtering to support real-time Skype detection," in *Proc. 6th Int. Wireless Communications and Mobile Computing Conf. (IWCMC '10)*, Caen, France, Jun./Jul. 2010, pp. 747–751.

[22] J.-L. Costeux, F. Guyard, and A.-M. Bustos, "Detection and comparison of RTP and Skype traffic and performance," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2006)*, San Francisco, CA, USA, Dec. 2006, pp. 1–5.

[23] L. Lu, J. Horton, R. Safavi-Naini, and W. Susilo, "Transport layer identification of Skype traffic," in *Proc. Int. Conf. Information Networking (ICOIN 2007)*, ser. LNCS, vol. 5200, Estoril, Portugal, Jan. 2007, pp. 465–481.

[24] S. Molnár and M. Perényi, "On the identification and analysis of Skype traffic," *Int. J. Commun. Syst.*, vol. 24, no. 1, pp. 94–117, Jan. 2011.

[25] K.-T. Chen and J.-K. Lou, "Rapid detection of constant-packet-rate flows," in *Proc. 3rd Int. Conf. Availability, Reliability and Security (ARES 08)*, Barcelona, Spain, Mar. 2008, pp. 212–220.

[26] L. Jun, Z. Shunyi, X. Ye, and S. Yanfei, "Identifying Skype traffic by random forest," in *Proc. Int. Conf. Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, Shanghai, China, Sep. 2007, pp. 2841–2844.

[27] P. A. Branch, A. Heyde, and G. J. Armitage, "Rapid identification of Skype traffic flows," in *Proc. 18th Int. Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV '09)*, Williamsburg, VA, USA, Jun. 2009, pp. 91–96.

[28] R. Alshammari and A. N. Zincir-Heywood, "Unveiling Skype encrypted tunnels using GP," in *Proc. IEEE Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, Jul. 2010, pp. 1–8.

[29] C.-C. Wu, K.-T. Chen, Y.-C. Chang, and C.-L. Lei, "Detecting VoIP traffic based on human conversation patterns," in *Proc. Principles, Systems and Applications of IP Telecommunications (IPTComm 2008)*, ser. LNCS, vol. 5310, Heidelberg, Germany, Jul. 2008, pp. 280–295.

[30] H. Zhang, Z. Gu, and Z. Tian, "Skype traffic identification based SVM using optimized feature set," in *Proc. Int. Conf. Information, Networking and Automation (ICINA 2010)*, vol. 2, Kunming, China, Oct. 2010, pp. 431–435.

[31] T. Yildirim and P. J. Radcliffe, "VoIP traffic classification in IPSec tunnels," in *Proc. Int. Conf. Electronics and Information Engineering (ICEIE 2010)*, vol. 1, Kyoto, Japan, Aug. 2010, pp. 151–157.

[32] B. Xu, M. Chen, C. Xing, and G. Zhang, "A network traffic identification method based on finite state machine," in *Proc. 5th Int. Conf. Wireless Communications, Networking and Mobile Computing (WiCom 2009)*, Beijing, China, Sep. 2009, pp. 1–4.

[33] N. M. Markovich and U. R. Krieger, "Statistical analysis and modeling of Skype VoIP flows," *Elsevier Comput. Commun.*, vol. 33, no. S1, pp. S11–S21, Nov. 2010.

[34] T. Okabe, T. Kitamura, and T. Shizuno, "Statistical traffic identification method based on flow-level behavior for fair VoIP service," in *Proc. 1st IEEE Workshop VoIP Management and Security (VoIP MaSe 2006)*, Vancouver, Canada, Apr. 2006, pp. 35–40.

[35] F. Liu, Z. Li, and J. Yu, "P2P applications identification based on the statistics analysis of packet length," in *Proc. Int. Symp. Information Engineering and Electronic Commerce (IEEC 2009)*, Ternopil, Ukraine, May 2009, pp. 160–163.

[36] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations," in *Proc. IEEE Symp. Security and Privacy (SP 2008)*, Oakland, CA, USA, May 2008, pp. 35–49.

[37] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical J.*, vol. 27, pp. 379–423, Jul. 1948.

[38] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A parameterizable methodology for Internet traffic flow profiling," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1481–1494, Oct. 1995.

[39] Tstat: TCP statistic and analysis tool. [Online]. Available: http://tstat.tlc.polito.it

[40] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*, 1st ed. Springer, Mar. 2008.

[41] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, Herndon, VA, USA, Feb. 1999, pp. 249–252.

[42] S. Lee, H. Kim, D. Barman, S. Lee, C. Kim, T. T. Kwon, and Y. Choi, "NeTraMark: A network traffic classification benchmark," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 22–30, Jan. 2011.

[43] L7-filter, application layer packet classifier for Linux. [Online]. Available: http://l7-filter.sourceforge.net

[44] Tools for L2-L7 traffic classification. [Online]. Available: http://netgroup.polito.it/research-projects/l7-traffic-classification/

[45] F. Risso, A. Baldini, and F. Bonomi, "Extending the NetPDL language to support traffic classification," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2007)*, Washington, DC, USA, Nov. 2007, pp. 22–27.

A Supplement to "Identification of Peer-to-Peer VoIP Sessions Using Entropy and Codec Properties"

João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

Abstract—This supplement is organized as follows. Appendix A presents the speech codecs analyzed in this work and describes how we studied the codecs used in the different versions of Skype. In appendix B, we explain the effect of the sliding window size in the entropy and present examples of the entropy analysis for different window sizes. Appendix C presents the summary of the packet length properties observed during the analysis of traffic from several VoIP sessions using different codecs. Appendix D characterizes the testbed used to collect datasets for the performance evaluation of the proposed method, describes the composition of the datasets, and presents additional details regarding the performance evaluation included in the main article. Furthermore, the results of the CPU time and memory used by the proposed classifier to process 13 distinct trace files are also included in appendix D.

Index Terms—Data communications, distributed applications, network communications, network management, network monitoring, packet-switching networks.

Appendix A
Studied Codecs

All the codecs used by more than one of the selected applications were studied in this work. In the case of *Skype*, all the codecs it supports were analyzed as they are mostly proprietary codecs that are not used by other applications.

A common codec used in Voice over Internet Protocol (VoIP) sessions is Pulse-Code Modulation (PCM), standardized in the *G.711* recommendation of the International Telecommunication Union (ITU). *G.711* defines two main compression algorithms, the μ-law algorithm (used primarily in North America and Japan) and the A-law algorithm (used in Europe and the rest of the world). The two versions of PCM are usually referred by the applications as PCMU and PCMA, respectively. For the sake of coherence, the same designations are used in the main article. *G.722* is based on Sub-Band Adaptive Differential Pulse Code Modulation (SB-ADPCM), which uses the baseline of PCM. *Speex* codec was also analyzed. It has three modes, ultra-wideband (sampling rate of 32 kHz), wideband (16 kHz), and narrowband (8 kHz) and supports Constant Bit Rate (CBR) and Variable Bit Rate (VBR).

Although the Global System for Mobile communications (GSM) is a standard for mobile telephone systems, it is commonly used to identify a speech codec. The original GSM speech codec is named Regular Pulse Excitation Long-Term Prediction (RPE-LTP). Nonetheless, as VoIP applications use the term GSM to refer to the codec, the same designation is used in this article.

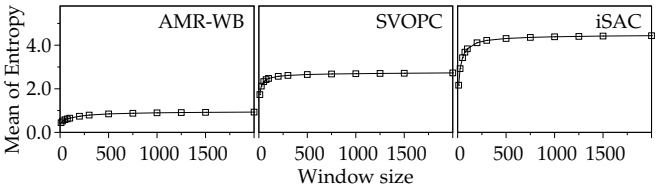

Fig. 1. Mean of the entropy of three examples of traffic for different sizes of the sliding window ranging from 10 to 2000 packets.

The Global IP Solutions (GIPS) company delivers embedded solutions used for communications. It offers a set of proprietary codecs that includes Internet Speech Audio Codec (iSAC), Internet Low Bit Rate Codec (iLBC), Internet Pulse Code Modulation wideband (iPCMwb), and Enhanced *G.711* (EG711) U/A. From these, only iLBC is royalty-free. In May 2010, GIPS was bought by *Google* and its codecs are now used by *Google Talk*.

All versions of the *Skype* software from 2.0 were tested to identify the different codecs supported throughout its evolution. The versions prior to 2.0 were not used as they seem to not be able anymore to connect to the authentication server of *Skype*. In order to know the codecs used by *Skype* in each version of the software, we started by identifying the codec used in a VoIP session by activating the *display technical call info* option in the *Skype* client. Afterwards, we used the *DisableCodecs* tag in the `config.xml` file to iteratively disable every

- *J. Gomes, P. Inácio, M. Pereira, and M. Freire are with Instituto de Telecomunicações, Department of Computer Science, University of Beira Interior, Portugal.*
  E-mail: jgomes@penhas.di.ubi.pt, {inacio, mpereira, mario}@di.ubi.pt
- *P. Monteiro is with Nokia Siemens Networks Portugal, S. A., with University of Aveiro, and with Instituto de Telecomunicações.*
  E-mail: paulo.1.monteiro@nsn.com

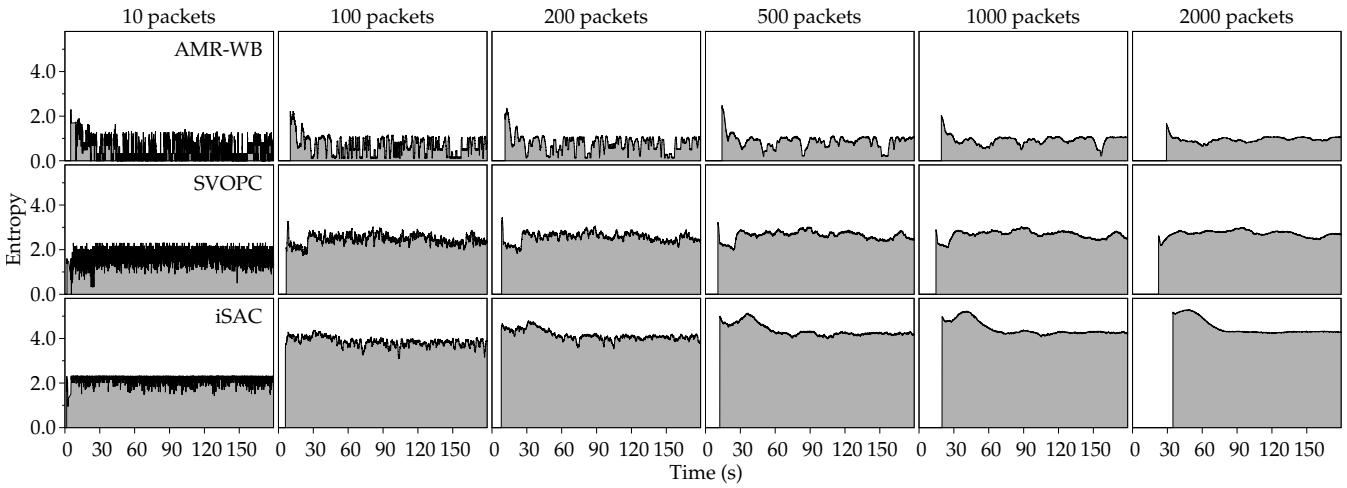

Fig. 2. Entropy analysis for three examples of traffic, using different sliding windows with sizes ranging from 10 to 2000 packets.

codec until no codec is left. We repeated this process for every *Skype* version and we concluded that, before version 3.0, *Skype* used *G.729*, PCM U/A, and the GIPS codecs. In version 3.0, it stopped using iPCMwb and the support for Adaptive Multi-Rate Wideband (AMR-WB) was added. From version 3.2 to 4.0, *Skype* used *G.729*, PCM U/A, AMR-WB and Sinusoidal Voice Over Packet Coder (SVOPC), a proprietary codec from *Skype*. In version 4.0, it stopped using AMR-WB and *SILK*, another *Skype* proprietary codec, was introduced. *SILK* has super-wideband (24 kHz) and wideband (WB) (16 kHz) modes and, since version 4.1 of *Skype*, it also has mediumband (MB) (12 kHz) and narrowband (NB) (8 kHz) modes. Another codec, identified by NWC in the `config.xml` file, is used since version 4.1 of *Skype*. Although in our analysis this codec presented properties similar to the ones from PCM, it was not possible to find any further information regarding this codec.

Appendix B
Sliding Window

The analysis of the entropy of the packet lengths described in the main article was performed for a sliding window with a constant size of N packets. Using a sliding window enables the real-time analysis during the duration of the flow, instead of analyzing the complete flow only when it has finished. We repeated the analysis for different window sizes, from 10 to 2000 packets, to understand the effect of the window size in the entropy value. Fig. 1 depicts the mean of the entropy using different window sizes, for three examples of traffic with distinct levels of entropy, low, medium, and high. In the figure, it is possible to observe that the entropy starts growing slower for a window size of 100 packets, and from the size of 500 packets, the entropy increases very slightly.

However, the effect of the sliding window size in the entropy value is of relative importance for the purpose of this work. Provided that the entropy is distinguishable for the different codecs, the absolute value of the entropy is not very relevant. More important is the stability of the entropy, because if the entropy varies significantly, passing the limits defined in several signatures, it would be difficult to make a classification decision. Fig. 2 presents the variation of the entropy using different window sizes. As the size increases, the entropy becomes more stable. However, it also takes more seconds to fill the window, depending on the number of packets per second generated by the codec. Hence, as a compromise between the stability of the entropy and the time it takes for the window to be filled, we chose to use sliding windows with size of 500 packets.

Appendix C
Properties of the Packet Lengths

In order to identify common characteristics in the packet lengths of the different codecs, we analyzed the traffic from several VoIP sessions made using different speech codecs. In this appendix, we present a summary of the results obtained in the analysis described in the main article.

Table 1 shows the frequent lengths observed for specific CBR codecs, as well as the interval in which the observed values of the entropy of the packet lengths are contained. The entropy the entropy of the packet lengths when using a CBR codec is almost always zero. However, for the *Skype* traffic, the entropy is always slightly higher.

In the case of VoIP sessions using VBR codecs, the packets lengths vary within a range of values. For this reason, the entropy is also higher when compared with the one obtained for the CBR codecs. The common values observed for VBR are presented in Table 2 for traffic of sessions from *Skype* or from Session Initiation Protocol (SIP) applications.

TABLE 1
Summary of the analysis of entropy and payload lengths of VoIP sessions using CBR codecs.

Codec	Entropy mean		Frequent lengths
	Incoming	Outgoing	(Byte)
		Skype	
PCMA	0.237	0.198	166, 168, 169, 170
PCMU	0.217	0.201	176, 178, 179, 180
G.729	0.215	0.202	26, 28, 29
iLBC	0.224	0.190	46, 47, 86
NWC	0.246	0.222	166, 169
		SIP applications	
PCMA	0.003	0.004	172
PCMU	0.002	0.004	172
G.722	0.000	0.001	172
GSM	0.003	0.005	45
iLBC	0.005	0.005	50, 88
Speex 32 kHz	0.001	0.000	49, 86
Speex 16 kHz	0.001	0.000	44, 82
Speex 8 kHz	0.001	0.000	32, 50

TABLE 2
Summary of the analysis of entropy and payload lengths of VoIP sessions using VBR codecs.

Codec	Entropy mean		Frequent lengths
	Incoming	Outgoing	(Byte)
		Skype	
EG711A	4.432	4.484	90–250, 200–400
EG711U	4.311	4.305	90–250, 200–400
iPCMwb	4.224	4.572	150–300, 300–600
iSAC	4.329	4.248	70–150, 170–300
SVOPC	2.468	2.413	15–120
SILK	3.676	3.651	40–120
SILK WB	3.704	3.607	40–120
SILK MB	3.020	2.9.24	30–60
SILK NB	2.868	2.828	20–50
AMR-WB	0.868	0.753	10–80
		SIP applications	
Speex 32 kHz	2.124	2.164	20–100
Speex 16 kHz	2.077	2.059	20–100
Speex 8 kHz	1.684	1.754	10–60

Appendix D
Performance Evaluation

In this appendix, we describe a few additional details regarding the performance evaluation of the proposed classifier.

D.1 Classification Evaluation

In order to avoid relying on the accuracy of a third-party classifier, we setup a testbed to capture the datasets used for the performance evaluation of the classifier, which is depicted in Fig. 3. This approach allowed us to be sure of which application generated each traffic flow. The testbed was formed by six groups of computers,

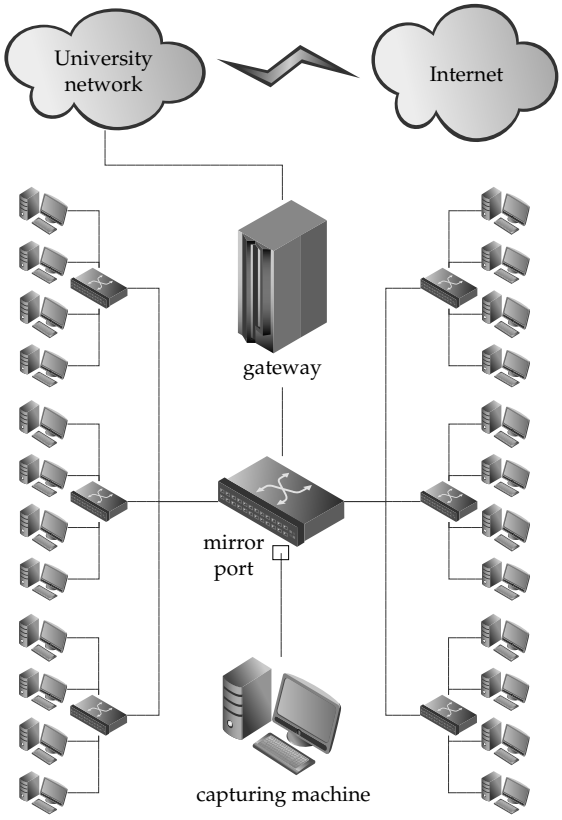

Fig. 3. Laboratory testbed in which the datasets used in the performance evaluation were captured.

TABLE 3
Datasets used to evaluate the performance of the classifier.

Dataset	Volume (GB)		Flows	
	TCP	UDP	TCP	UDP
Dataset 1	1.3	0.4	7454	1007
Dataset 2	1.0	0.6	9053	1185
Dataset 3	2.3	0.8	10307	1910
Dataset 4	5.5	10.0	31933	42114

each of the groups was connected to a switch, and the six switches were connected to a different switch, which connects the Local Area Network (LAN) to the gateway. This enables the connection to the Internet through the university network. Three groups were running the *Microsoft Windows* operating system, while the other three groups were running *Linux*. The application running in the computers vary in each dataset, and the VoIP sessions were made from different computers in the testbed to *Windows* and *Linux* computers or *Android* smartphones outside or in another LAN in the university network. The traffic was captured from a mirror port in the main switch.

Using the testbed, we collected traffic from VoIP sessions and kept a record of the applications running in each machine. The captured traffic is divided into four

TABLE 4
Composition of the datasets used in the performance evaluation.

Traffic	Dataset 1				Dataset 2				Dataset 3				Dataset 4			
	Bytes (%)		Flows (%)		Bytes (%)		Flows (%)		Bytes (%)		Flows (%)		Bytes (%)		Flows (%)	
	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP
HTTP download	04.09	00.12	09.71	01.44	05.78	00.65	03.29	00.37	00.39	00.03	04.33	00.82	06.77	00.00	00.08	00.09
Web browsing	01.05	00.00	04.12	00.58	00.57	00.03	04.69	00.37	00.26	00.02	02.89	00.55	00.07	00.01	00.82	00.11
Streaming	53.78	18.19	40.87	03.47	49.16	30.36	51.28	05.82	61.57	20.12	50.30	05.58	01.74	12.03	03.07	03.18
Telnet / SSH	00.44	00.01	01.04	00.15	00.21	00.01	01.76	00.14	00.00	00.00	00.00	00.00	00.04	00.01	00.20	00.11
FTP / SFTP	00.93	00.03	02.20	00.33	00.64	00.04	05.27	00.42	00.00	00.00	00.00	00.00	04.51	00.00	00.11	00.05
P2P streaming	03.16	01.07	09.43	00.80	01.74	00.98	07.89	00.90	04.35	01.42	06.71	01.00	16.58	35.39	12.34	23.90
P2P file-sharing	06.33	02.14	12.58	01.07	01.16	00.65	03.29	00.37	06.52	02.13	10.06	01.40	04.73	16.92	14.55	37.71
VoIP	05.41	03.24	08.12	04.08	03.25	04.77	10.59	03.56	01.35	01.86	10.01	06.36	01.08	00.12	02.54	01.14

TABLE 5
Codecs used for the VoIP sessions included in the performance evaluation datasets.

Traffic	Dataset 1				Dataset 2				Dataset 3				Dataset 4			
	Bytes (%)		Flows (%)		Bytes (%)		Flows (%)		Bytes (%)		Flows (%)		Bytes (%)		Flows (%)	
	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP	TCP	UDP
Skype AMR-WB	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	03.68	03.63	01.75	07.62
Skype G.729	02.51	01.48	01.34	00.22	02.09	03.56	07.91	04.85	00.00	00.00	00.00	00.00	04.51	00.29	01.94	03.22
Skype PCMA	06.03	04.26	01.55	00.31	06.07	09.92	02.84	06.72	00.00	00.00	00.00	00.00	08.20	00.30	01.61	02.65
Skype PCMU	06.89	04.59	01.65	00.10	04.97	09.75	03.36	06.04	00.00	00.00	00.00	00.00	07.88	00.28	01.47	02.46
Skype NWC	04.19	04.38	02.07	01.24	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	03.29	00.12	00.90	01.84
Skype iSAC	00.00	00.00	00.00	00.00	01.61	03.02	01.34	02.46	00.00	00.00	00.00	00.00	02.88	00.34	00.28	02.22
Skype iPCMwb	00.00	00.00	00.00	00.00	07.22	09.06	01.79	03.06	00.00	00.00	00.00	00.00	09.12	05.80	00.90	05.30
Skype EG711A	00.00	00.00	00.00	00.00	05.84	06.85	02.09	03.43	00.00	00.00	00.00	00.00	05.47	00.34	00.38	02.41
Skype EG711U	00.00	00.00	00.00	00.00	07.11	08.25	01.64	03.06	00.00	00.00	00.00	00.00	06.11	00.34	00.38	02.51
Skype iLBC	00.00	00.00	00.00	00.00	02.59	02.71	02.01	03.58	00.00	05.13	00.09	02.05	02.07	00.35	01.51	03.17
Skype SILK	07.79	05.18	03.51	00.31	00.00	00.00	00.00	00.00	00.00	06.25	00.11	02.49	09.64	00.57	06.24	07.14
Skype SILK WB	07.75	04.71	03.20	00.37	00.02	01.86	01.04	01.42	00.00	00.00	00.00	00.00	08.27	00.41	01.84	03.41
Skype SILK MB	04.39	02.01	01.45	00.21	00.04	01.61	01.79	01.79	00.00	00.00	00.00	00.00	02.08	00.12	00.19	00.80
Skype SILK NB	04.59	01.61	01.65	00.10	00.04	00.72	02.39	01.64	00.00	00.00	00.00	00.00	02.02	00.11	01.70	01.47
Skype SVOPC	07.37	08.36	03.41	01.96	00.00	00.00	00.00	00.00	00.00	04.01	00.00	00.00	05.47	00.24	00.80	02.70
SIP PCMA	01.25	00.16	00.27	00.08	00.00	00.30	00.04	00.23	00.01	09.33	00.16	03.72	00.00	00.00	00.00	00.00
SIP PCMU	00.00	00.00	00.00	00.00	00.00	00.37	00.05	00.28	00.14	09.21	00.93	06.20	00.00	00.00	00.00	00.00
SIP *G.722*	01.56	00.10	00.34	00.04	00.00	00.00	00.00	00.00	00.01	11.66	00.21	04.65	00.00	00.00	00.00	00.00
SIP GSM	00.00	00.00	00.00	00.00	00.00	00.34	00.03	00.26	00.00	07.27	00.07	05.64	00.00	00.00	00.00	00.00
SIP iLBC	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.01	13.99	00.25	05.58	00.05	00.57	00.03	02.86
SIP *Speex* 32 kHz	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.22	07.46	01.40	05.11	00.07	00.71	00.04	03.07
SIP *Speex* 16 kHz	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.01	09.21	00.14	09.00	00.00	00.00	00.00	00.00
SIP *Speex* 8 kHz	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.01	07.18	00.28	09.13	00.00	00.00	00.00	00.00

datasets with distinct sizes as presented in Table 3, corresponding to different capturing periods, from January to March 2011. Besides of the VoIP sessions, each dataset also contains traffic from other classes of applications, such as web browsing (excluding streaming contents, which are included in the streaming class), Hypertext Transfer Protocol (HTTP) downloads (download of a large file, e.g., a disc image or an executable), file transfer (File Transfer Protocol (FTP) and Secure File Transfer Protocol (SFTP)), remote sessions (*Telnet* and Secure Shell (SSH)), live or on-demand audio and video streaming (Real-Time Streaming Protocol (RTSP), HTTP, Microsoft Media Server (MMS), and *Flash*), Peer-to-peer (P2P) video streaming (*PPStream*, *TVU Player*, and *SopCast*) and P2P file-sharing (*eDonkey*, *BitTorrent*, and *Gnutella*), so as to maximize the possibility of having false positive cases. Table 4 presents the share of the different classes of applications in each dataset.

TABLE 6
Average time (s) needed by the classifier to correctly classify VoIP sessions in the first and second classifications, using the signatures of the bit rate, group, and codec levels.

Traffic	Bit rate level		Group level		Codec level	
	First	Second	First	Second	First	Second
UDP	9	33	12	–	11	39
TCP	14	–	14	–	12	–
All	11	33	13	–	12	39

The VoIP sessions included in the datasets were made

TABLE 7
Percentage of false positives in the traffic from each class of non-VoIP applications, using the signatures of the bit rate, group, and codec levels.

Traffic	Dataset 1			Dataset 2			Dataset 3			Dataset 4		
	Bit rate	Group	Codec	Bit rate	Group	Codec	Bit rate	Group	Codec	Bit rate	Group	Codec
HTTP download	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00
Web browsing	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00
Streaming	00.05%	00.00%	00.00%	00.01%	00.00%	00.00%	00.03%	00.00%	00.00%	00.11%	00.06%	00.03
Telnet / SSH	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00
FTP / SFTP	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00
P2P streaming	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	01.64%	01.51%	00.01
P2P file-sharing	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.00%	00.04%	00.00%	00.00

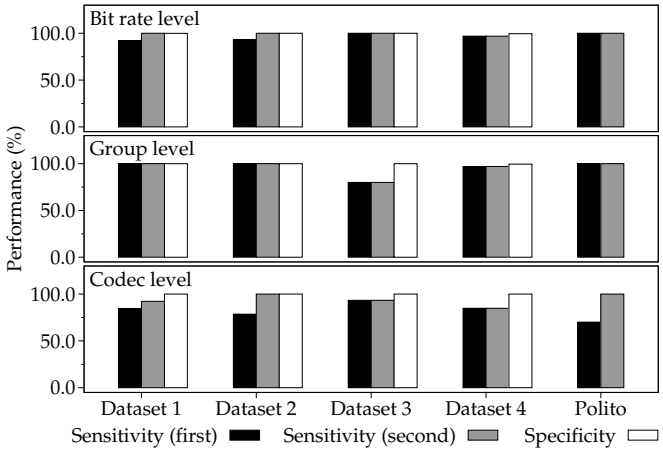

Fig. 4. Performance results of the proposed classifier for the different levels of signatures.

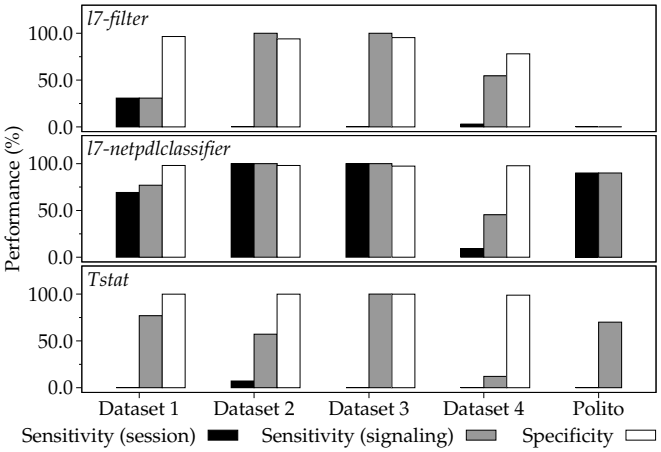

Fig. 5. Performance results of the other tested classifiers.

using all the applications and speech codecs described in the main article. Table 5 presents the share of VoIP traffic in each dataset for the analyzed speech codecs, with the exception of the data used by the applications to authentication and contact status synchronization. Datasets 1 and 2 contain a greater percentage of *Skype* traffic, while dataset 3 includes more sessions from SIP applications. Dataset 4 includes primarily *Skype* sessions over the Transmission Control Protocol (TCP).

The performance of the proposed classifier and of three other available classifiers was evaluated using datasets 1, 2, 3, and 4, as described in the main article. The classifier continually analyzes every packet since the beginning of the flow until its end. In every moment, if the traffic is matched by one of the signatures, the flow is classified as being generated by a VoIP session using the corresponding codec. In some cases, the first classification produced by the classifier is not correct. Nevertheless, since the classifier continues to analyze every packet, if in any moment the signature does not match the traffic anymore but a different signature does, the classifier corrects the classification. For this reason, the performance evaluation presented in the main article includes the results for the first and second (if there is a second one) classifications. Figs. 4 and 5 present a graphical representation of the results included in Tables 3 and 4 of the main article. Additionally, the average time that the classifier took to correctly classify the VoIP sessions is included in Table 6. During the performance evaluation, the false positive cases were caused by the traffic from streaming, P2P file-sharing, and P2P streaming applications. Table 7 presents the percentage of flows in the traffic from each class of applications that were misclassified, for all the signature levels.

D.2 Computational Resources Analysis

In order to understand how the computational resources consumption grows as the traffic increases, we measured the CPU time and memory used by the classifier to process trace files with different amounts of packets and flows. As described in the main article, the CPU time used by the classifier depends on the number of processed packets whose transport layer payload is larger than 5 bytes, whereas the used memory depends on the number of processed flows containing packets with payload larger than 5 bytes. The measurement results are listed in Table 8 and show the linear dependence on the number of flows and packets with payload larger than 5 bytes.

TABLE 8
Measurements of CPU time and maximum memory used by the classifier to analyze 13 distinct trace files and their dependence on the number of packets whose payload is larger than 5 bytes and on the number of flows containing packets whose payload is larger than 5 bytes, for the signatures of bit rate, group, and codec levels.

Trace files	**Total**		**Larger 5 bytes**		**Bit rate level**		**Group level**		**Codec level**	
	Packets	Flows	Packets	Flows	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)
Trace file 1	269322	411	226497	219	17.32	11504	19.25	11472	93.80	11584
Trace file 2	773036	2152	598842	869	47.43	20368	51.79	20320	251.27	20416
Trace file 3	855554	995	680915	452	55.09	14528	61.57	14512	295.31	14592
Trace file 4	1039562	3481	736894	802	58.70	18896	65.24	18848	309.95	18944
Trace file 5	1337017	2365	1256111	1499	102.49	26432	107.79	26336	482.77	26336
Trace file 6	1736251	2997	1635476	1813	131.39	29968	142.78	29856	642.20	30192
Trace file 7	2333035	3815	2203802	2251	181.42	34768	197.03	34656	892.06	35056
Trace file 8	3031885	4765	2869311	2782	240.97	40432	260.71	40320	1180.06	40736
Trace file 9	3833565	9773	3184102	6350	271.95	72288	293.56	72112	1318.57	72720
Trace file 10	4033409	10123	3342619	6555	287.22	74208	309.92	74032	1394.13	74640
Trace file 11	3537662	5533	3349693	3189	286.42	44960	306.77	44816	1383.81	45280
Trace file 12	4229292	10666	3510875	6910	295.13	77680	324.11	77504	1465.09	78160
Trace file 13	4408424	10967	3658008	7074	305.55	79168	338.80	78992	1517.03	79632

Chapter 6

Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy

This chapter consists of the following article:

Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy

João V. Gomes, Pedro. R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

Article submitted for publication in an international journal.

Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy

João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

Abstract—The identification of network traffic generated by a given application constitutes an important asset for the management of computer networks. Nonetheless, the popularity of the peer-to-peer (P2P) paradigm, the growth of the throughput of computer networks, and the use of payload encryption increased the complexity of traffic classification and reduced the effectiveness of Deep Packet Inspection (DPI) approaches. In this article, we propose a novel classification mechanism based mostly on the packet lengths, which can be applied to encrypted traffic since it does not use payload data. The heterogeneity of the packet lengths from several applications is analyzed considering different perspectives, namely subsets of the traffic containing the packets of both directions or from different ranges of packet lengths. Additionally, the heterogeneity of inter-arrival times and remote *host/port* pairs is used to improve the results in specific cases. We resort to the mean of entropy to define classification rules for P2P streaming and file-sharing flows. In order to make the method suitable for real-time operation, the entropy values are computed for a sliding window with a constant size of *N* packets. It is shown that the performance of the proposed method achieves an accuracy greater than 95%.

Index Terms—Data communications, distributed applications, network communications, network management, network monitoring, packet-switching networks.

1 Introduction

The effective management of computer networks comprises several responsibilities, including the organization and design of the network, the balance of the traffic load, the capacity to assure the Quality of Service (QoS) required by applications with distinct priority levels, or the implementation of security measures for different applications. In such context, traffic classification arises as a crucial network management tool to retrieve the nature of the traffic, by enabling the identification of the application or service responsible for each traffic flow [1], [2], [3].

The classification of Internet traffic started as a simple operation based on the port numbers used by the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) [4], [5], [6]. Since most applications or services used well-known port numbers for their connections, e.g., port 80 for Hypertext Transfer Protocol (HTTP) or port 21 for File Transfer Protocol (FTP), it was straightforward to identify the application associated with a certain flow by simply looking at the port number used in the connection. However, some applications were subject to rigorous traffic shaping schemes or even completely blocked by Internet Service Providers (ISPs) and network administrators, mostly because they were heavy bandwidth consumers. Therefore, many of them, especially peer-to-peer (P2P) software clients, started to use random port numbers or well-known ports used by other protocols, e.g., port 80, rendering port numbers useless as a classification solution [7], [8].

In order to overcome the inaccuracy of the port-based methods, different Deep Packet Inspection (DPI) mechanisms were proposed for traffic classification [7], [9], [10], [11]. Generally, DPI methods rely on a database of signatures associated with specific applications. The signatures are formed by byte strings present in the payload of packets from the target application. The classifier tries to match each signature with the contents in the data field of the packets. DPI methods are typically very accurate as they inspect the data carried within the packet. Nonetheless, besides requiring previous knowledge of the target application, the deep inspection of every packet may be a computationally demanding operation in high-speed networks. This fact is exacerbated by the proliferation of new Internet applications, which implies a larger number of signatures in the database that must be checked with each packet. In addition, since they are based on the contents of the packets, DPI mechanisms may also raise privacy issues [12]. Furthermore, many applications are adopting evasive measures, as protocol obfuscation and payload encryption, that may render DPI methods ineffective as they prevent them from using the payload data [13], [14].

The constraints of the DPI approach motivated the development of alternative methods. Several authors have been proposing *in the dark* classification mechanisms [15], [16], which do not resort to the payload data. Generally, these methods are based on the behavior of the protocol or application and are implemented following a few distinct approaches. In some studies, the classification

- *J. Gomes, P. Inácio, M. Pereira, and M. Freire are with Instituto de Telecomunicações, Department of Computer Science, University of Beira Interior, Portugal.*
  E-mail: jgomes@penhas.di.ubi.pt, {inacio, mpereira, mario}@di.ubi.pt
- *P. Monteiro is with Nokia Siemens Networks Portugal, S. A., with University of Aveiro, and with Instituto de Telecomunicações.*
  E-mail: paulo.1.monteiro@nsn.com

Manuscript received day month 2011.

relies on a set of heuristics that are usually focused on behavioral characteristics of the protocol, e.g., the concurrent use of TCP and UDP, or the existence of multiple connections between the same hosts [17], [18]. In other cases, the authors have explored different packet-level or flow-level traffic properties, like inter-arrival times, packet lengths, or flow durations, by means of statistical analysis [14], [19] or machine learning algorithms [20], [21]. Although *in the dark* classification has typically lower accuracy than DPI, it offers a good compromise between effectiveness and computational cost. Furthermore, its accuracy is not affected by the encryption of the traffic. Nevertheless, the search for improving the effectiveness of *in the dark* classification is leading to the increase of the complexity of this kind of methods. Cascarano et al. [22] compared the performance of a DPI classifier and a Support Vector Machine (SVM)-based method and concluded that both have similar computational cost.

Due to the limitations of the different approaches and in face of its importance for network management, traffic classification represents a very active research topic. Additionally, the rise of new Internet paradigms that shift the focus to the user, like P2P, reinforced its significance. The traffic of this kind of applications presents properties distinct from the applications relying on the traditional *client-server* paradigm. Moreover, their greedy nature, in conjunction with the increase of the network throughput, raised the workload in computer networks. Additionally, the growing popularity of P2P systems has also motivated discussions regarding the security issues that may be inherent to these applications. A few authors have analyzed the vulnerabilities of P2P networks [23], [24] and concluded that they represent a real security risk for individual users and organizations. These reasons, along with the the increasing use of evasive techniques in P2P applications, make these systems particularly challenging for traffic classification. The large number of articles on the classification of P2P traffic (specifically) is an evidence of the attention given by the researchers to the identification of this particular class of traffic [25], [26], [27], [28], [29], [30], [31].

This study aims to propose a classifier that does not rely on payload data, is capable of identifying flows from P2P applications, and is suitable to operate in real-time. In a previous work [32], we made a preliminary study on the lengths of the packets from P2P and non-P2P applications. The obtained results showed that the packets generated by P2P applications presented more varied lengths. This heterogeneity is either caused by the aggregation of multiples flows with distinct properties used to communicate with several peers, or by the speech codec used by P2P Voice over Internet Protocol (VoIP) applications. On the other hand, the traffic from non-P2P applications was mostly formed my a single or a few flows, resulting in packets with homogeneous lengths. In two previous works, we studied a large set of speech codecs used by P2P VoIP applications and proposed a classifier for VoIP flows [33], and we also proposed a classifier to identify aggregated traffic from hosts running P2P applications [34]. In both cases, the level of heterogeneity of the lengths of the packets was measured using entropy. Nevertheless, P2P file-sharing and P2P video streaming applications use one-to-many connections, as they share a file or a multimedia stream with many peers. In order to identify individual flows from one-to-may P2P applications, it is necessary to separate the traffic into flows. This separation destroys the main cause of the heterogeneity of the lengths of the packets explored in these previous works. Therefore, in order to tackle the classification of these specific classes, we had to follow a distinct approach described herein.

In this article, a method to identify, in real-time, the flows generated by one-to-many P2P applications is proposed. The mechanism does not resort to the payload data, making it applicable to encrypted traffic. In order to decrease the complexity, the classifier is mostly based on only one traffic feature, the packet length. Additionally, to improve the accuracy in some cases, the inter-arrival times and the variety of remote *host/port* pairs were also used. The method is based on the analysis of the heterogeneity of the lengths of the packets from different dimensions of the traffic, being that heterogeneity measured through entropy. The entropy value is computed using a sliding window with size of N packets so that the analysis could be performed in real-time. To the best of our knowledge, this is the first flow classification method exploring the heterogeneity of the lengths (instead of the individual values or their distribution) and using entropy to measure it. The performance of the classifier was evaluated using datasets with ground truth information carefully obtained in a testbed. The results show that the proposed method is accurate.

The remainder of the article is organized as follows. Section 2 contains a description of the previously published related work. Section 3 analyzes the properties of the packets from different types of applications and explains how entropy may be used to measure their heterogeneity. Section 4 presents the entropy-based classifier, followed by its performance evaluation in section 5. The article finishes with a section summarizing the most important conclusions.

2 Related Work

The early classification scheme based on port numbers was used by a few preliminary studies on the measurement of the P2P data in computer networks [4], [5], [6]. More recently, the port numbers were used mostly in cooperation with other classification approaches, as DPI [35] or statistics [36].

Due to the low accuracy of the port-based methods, many researchers relied on DPI to implement classification mechanisms. Generally, DPI techniques are based on byte strings found in the packet payload. Methods following this approach use those strings as signatures

associated with specific applications [7], [37]. Although payload encryption usually renders DPI useless, in [11] the authors were able to successfully classify traffic flows from a P2P application that uses encryption, by matching payload signatures in the first packets. In other studies [29], [38], [39], [40], the authors proposed mechanisms to automatically extract signatures from the packet payload.

DPI-based classifiers are generally the most accurate ones. Nevertheless, they also require, in most cases, more computation resources. Hence, several studies proposed lightweight DPI implementations. Risso et al. [9] analyzed a few efficient DPI classifiers. Kumar et al. [41] and Smith et al. [10] used a deterministic finite automata (DFA) to optimize the process of signature matching, while Liu et al. [42] resorted to chip multiprocessors (CMPs) to implement a high-speed and memory-efficient method. Cascarano et al. [43] proposed two optimizations for a DPI classifier that significantly increases the efficiency of the mechanism at the cost of a small decrease of its accuracy.

Some authors are also using payload data for traffic classification in alternative ways. Park et al. [44] proposed a classification method, based on payload data, that does not resort to byte strings. Instead of defining signatures for each target application, they explored common behavior patterns of different tasks of each application. Using an algebraic model, they converted a number of bytes in the payload to a vector and employed *Jaccard* similarity as a distance metric to compare the models. Dorfinger et al. [45] resorted to entropy to measure the heterogeneity of the payload at the byte level, whereas Bonfiglio et al. [46] used the *Chi-squared* test. Xu et al. [28] compared the contents of the received and sent data to identify P2P traffic.

In order to avoid the drawbacks of DPI, many authors have proposed statistical methods based on packet or flow features [47], [48], [49], [50]. Dusi et al. [14] used statistical analysis to identify traffic in encrypted tunnels, while in [51], the authors proposed a statistical method to distinguish between VoIP and Web traffic. Palmieri and Fiori [52] presented a mechanism for traffic classification based on recurrence quantification analysis. A few studies [53], [54], [55] have also proposed methods relying on the calculus of packet length distributions, using different distance metrics to choose the best match. Alternatively, some authors used heuristics [17], [18], [30] to model the behavior of applications, while in [15] and [56], the authors explored the *social* relationship between hosts in different ways and identified patterns to be used in traffic classification.

More recently, numerous studies have proposed classification methods based on different machine learning algorithms. A few authors have based their approaches on clustering algorithms, such as *k-means* [20], [57], [58], [59], expectation-maximization algorithm [60], [61], gaussian mixture model and spectral clustering [62], or minimum spanning tree clustering [63]. Some studies proposed classification methods resorting to decision trees, as *C4.5* [64], Very Fast Decision Tree (VFDT) [65], [66], [67], Classification And Regression Tree (CART) [19], or *random forest* [68]. Other works have used *hidden Markov* models [69], [70], Naïve Bayes and neural networks [27], [71], [72], [73], or *k*-nearest neighbor [74], [75]. In the last few years, many authors [21], [31], [76], [77], [78], [79], [80] have been presenting methods based on SVMs, using a variety of traffic features.

Several studies [81], [82], [83], [84], [85], [86], [87], [88], [89] have also combined a set of different machine learning algorithms from Naïve Bayes and neural networks, SVMs, *C4.5*, *k-means*, *random forest*, expectation-maximization, *k*-nearest neighbor, and *AdaBoost*. More complex classifiers have been proposed [25], [36], [90], [91], [92], [93], associating distinct approaches from port numbers, DPI, statistics, heuristics, and different machine learning algorithms.

In order to avoid the increasing complexity of recent payload-agnostic methods, the classification mechanism described herein has the intention of minimizing the number of traffic features used in the process. Hence, the proposed mechanism uses mostly the packet lengths to classify the traffic. Although several studies on traffic classification have already employed the lengths of the packets, they used them in association with other traffic features, by calculating statistics like the mean or the variance, or by estimating the probability distributions. On the contrary, this method explores the variety of values in a set of packet lengths and it resorts to entropy to measure the level of heterogeneity of those values. We used this approach, for the first time, in a preliminary study [32] that compared the characteristics of the lengths of the packets for P2P and non-P2P traffic. In the case of P2P file-sharing, these evidences result from the aggregation of multiple and distinct flows, while, for VoIP flows, it is consequence of the Variable Bit Rate (VBR) speech codecs used by many popular VoIP applications. Based on this behavior, in [33], we implemented a classifier for VoIP flows which relies on the codec used in the session. In addition, in [34], we proposed a different classifier that identifies hosts running P2P applications, by exploring the heterogeneity caused by the aggregation of flows. In the sequence of these two works, we now address the classification of individual flows from P2P video streaming and P2P file sharing, which is especially challenging due to the separation of flows and consequent reduction of the heterogeneity of the packet lengths.

The most similar work to our study herein described is the one recently published by Li et al. [94]. They followed a similar approach to distinguish Constant Bit Rate (CBR) and VBR VoIP traffic. Nonetheless, the method they proposed operates offline, only for the entire flows, and was not applied to other classes of traffic. Furthermore, instead of classifying entire flows offline, in time intervals, or only for the first packets as most studies do, the mechanism described herein implements

a sliding window with size of N packets. The use of the window enables the real-time and efficient operation and makes it possible to analyze every variation in the traffic during the lifetime of the flow. Unlike most previous works, we analyzed the traffic of several applications from different types of protocols, and we compared the results obtained for P2P traffic with the ones obtained for several distinct non-P2P applications.

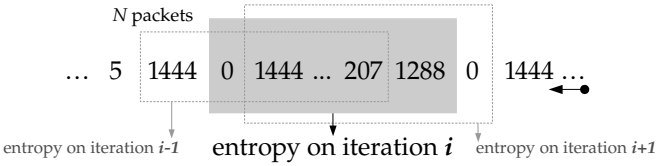

Fig. 1. Representation of the use of a sliding window with constant size of N packets to calculate the entropy for the packet lengths.

3 Properties of the Packet Lengths in Flows

The study presented herein, as well as the proposed classifier, are primarily based on the analysis of the packet lengths, though other traffic features were also considered. This section describes the datasets analyzed in this work, explains how entropy was used to measure the heterogeneity of the traffic features, and describes the observed behavior.

3.1 Applications and Datasets

In order to study the characteristics of the packets from different applications, we collected network data directly on end user machines running a single application. This approach allows us to be sure of the ground truth information of the datasets without relying on the accuracy of a third-party classifier.

Given the purpose of this work, we captured the data generated by several sessions of different kinds of P2P systems, namely file-sharing, video streaming, and VoIP. Since the properties of VoIP packets depend on the used speech codec, this type of traffic requires a detailed study so as to consider the influence of different codecs. Hence, on a previous work, we addressed this particular case analyzing several codecs and VoIP applications and proposed a P2P VoIP traffic classifier. Nonetheless, to provide a more complete analysis here, we also collected traffic from popular P2P VoIP applications with the default speech codec configurations. Additionally, we captured data generated by several non-P2P applications that enables a comparison with the properties of P2P traffic and the identification of dissimilarities and classification patterns.

The datasets were captured between October 2006 and July 2011 on different end hosts running *Linux* and *Microsoft Windows* operating systems, in different connection scenarios (small and large Local Area Networks (LANs) and commercial home links). We selected several types of applications and a few protocols for each of them. In the case of P2P file-sharing, to address the possibility of some applications having slightly distinct implementations of the protocol, we used different popular applications for each protocol, e.g., *BitTorrent*, *Transmission*, and *Vuze* (*BitTorrent* network), *Gnutella*, *Frostwire*, and *Limewire* (*Gnutella* network), *eMule* and *aMule* (*eDonkey* network). The collected data totals 10.76 GB, shared between different types of traffic. In the supplemental material, we included a table (Table 1) which presents the percentage of each type of traffic in the experimental data. The following protocols or applications were used in the datasets:

- P2P file-sharing – *BitTorrent*, *eDonkey*, and *Gnutella*;
- P2P streaming – *SopCast*, *PPStream*, and *TVU Player*;
- P2P VoIP – *Skype* and *Google Talk*;
- Mail – Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and Internet Message Access Protocol (IMAP);
- HTTP download – the download of a non Hypertext Markup Language (HTML) file, e.g., an executable or a disc image;
- Web browsing – web page browsing (excluding the streaming of multimedia contents, which is considered in the next class);
- Streaming – on-demand and live, audio and video, Microsoft Media Server (MMS), Real-Time Streaming Protocol (RTSP), HTTP, and *Flash*-based streaming;
- Remote shell – *Telnet* and Secure Shell (SSH);
- File transfer – FTP and Secure File Transfer Protocol (SFTP);
- Online gaming – *RuneScape* and *War of Legends*.

3.2 Evaluation of Entropy Using a Sliding Window

The work presented herein is based on the analysis of the heterogeneity of a reduced set of traffic features. In order to assess the level of heterogeneity for each of them, we resorted to the concept of entropy described in the information theory by Shannon [95]. Shannon introduced entropy as measure of the uncertainty of a random variate, defined by

$$H(x) = -\sum_{i=1}^{n} p(x_i) \ln p(x_i), \tag{1}$$

where n and $p(x_i)$ represent the number of possible occurrences of x and the probability of the particular occurrence of x_i, respectively. For a finite number $n \in \mathbb{N}$, the maximum value $H(x)$ may attain is given by

$$H^{max}(x) = \ln n. \tag{2}$$

Entropy is always a positive number. $H(x)$ is close to 0 when the pool of samples is extremely homogeneous and increases with the number of distinct occurrences under analysis.

Instead of computing the entropy of each feature for an entire flow or a large amount of traffic, we defined a

window containing a fixed number of packets. We then slid the window through the traffic and computed the entropy of one of the features for all the packets within the window. For each distinct traffic feature, we used a different window. Fig. 1 depicts the entropy evaluation using a sliding window for the packet length feature. In each iteration of the window, the oldest packet leaves the window, a new one is added, and a new entropy value is obtained. This process enables a continuously updated evaluation of the entropy, through the entire lifetime of a flow, starting immediately after the first N packets (needed to fill the window). The ability to obtain a new entropy value every time a new packet arrives enables the real-time analysis of the traffic.

Nevertheless, obtaining the entropy of N values every time a new packet arrives requires the calculation of the probabilities of the values for all the packets within the window in each iteration. This may be a demanding process, especially when the values are more varied. Therefore, we optimized the entropy evaluation by defining it recursively in relation to the entropy obtained in the previous iteration. When a packet leaves the window and another one is added, only a maximum of two probabilities change. Hence, we use (1) to calculate the entropy only the first time the sliding window is filled. After that, instead of calculating the probabilities of the analyzed feature of all the packets in the sliding window, we compute only the ones that concern the packet that leaves and the packet that is added to the window. We update their influence in the entropy using

$$U(x) = p_{i-1}(x) \ln p_{i-1}(x) - p_i(x) \ln p_i(x), \tag{3}$$

$$H_i(x) = H_{i-1}(x) + U(x_o) + U(x_l), \tag{4}$$

where $p_{i-1}(x)$ and $p_i(x)$ are the probability of feature x in iterations i and $i-1$ of the sliding window, respectively, and x_o and x_l are the value of feature x in the oldest packet in window and in the packet that will be added, respectively.

3.3 Level of Analysis

Since our main purpose is to identify P2P traffic, we looked for characteristics in the packets that could be distinguishable from the ones generated by non-P2P traffic. In a previous work [34], we compared the packet lengths from aggregated traffic generated by individual hosts running P2P or non-P2P applications. The analysis showed that the packet lengths from P2P traffic are more heterogeneous. This pattern allowed us to identify hosts running P2P applications. However, the level of heterogeneity of P2P streaming and file-sharing traffic is mostly caused by the aggregation of multiple flows between different peers. Hence, the same approach is not enough to classify the individual flows of each host that result from a P2P application.

Therefore, we used the traffic described in section 3.1 to study the characteristics of the packet lengths from the flows of different types of Internet applications. Although our main goal was to identify the individual flows generated by P2P applications, we decided to separate the traffic based only on the host Internet Protocol (IP) address and TCP or UDP port number. This approach allows us to explore the fact that some applications establish several connections in the same port number. Many P2P applications receive requests in the same port number, which results in several flows using the same port. Thus, it is possible to analyze all the traffic from one port number, without separating it by the flows that use that port, and classify them as being generated by the same application. Furthermore, from the packet lengths we excluded the bytes concerning the *Ethernet*, IP, TCP, and UDP headers. Hence, we used only the lengths of the TCP or UDP payloads.

In [34], we analyzed the entropy using different window sizes from 10 to 2000 packets and we observed that, for larger sliding windows, the entropy is more stable through the iterations. However, a larger window needs more packets to produce the first entropy value. Hence, as a compromise between the two factors, we chose to use here sliding windows with size of 100 packets.

3.4 Heterogeneity of the Packet Lengths

In order to study the heterogeneity of the packet lengths, we analyzed the entropy separately for each host address and TCP or UDP port number belonging to the network being monitored. The examples included in the figures of this section are not meant to be a representation of the traffic from all the *host/port* pairs of each of the depicted applications. Our purpose was to illustrate distinct traffic patterns that we observed in the analyzed datasets.

Figs. 2, 3, and 4 depict the cumulative probability distributions of the lengths of the first 10000 packets of a few examples of flows from non-P2P, P2P file-sharing, and P2P video streaming applications. We chose to represent the cumulative probability distribution so that it could be easier to observe the density of packets with a certain length. In the same figures, we included the corresponding entropy, which was calculated using the method described in section 3.2 for a window with a size of 100 packets. Since this analysis is applied to traffic classification, the stability of the entropy is important to assure that the results are not affected by occasional entropy variations. Therefore, we used, in each iteration, the mean of the entropy in all the steps of the window since the beginning of the flow to the present iteration. The calculation of the mean was implemented in real-time using

$$\mu_i = \frac{H_i(x) + (i-1)\mu_{i-1}}{i}, \tag{5}$$

where μ_i denotes the mean of the entropy for all the iterations until i. Nevertheless, in appendix B of the supplemental material, we included plots similar to the ones in Figs. 2, 3, and 4 that depict the lengths of the

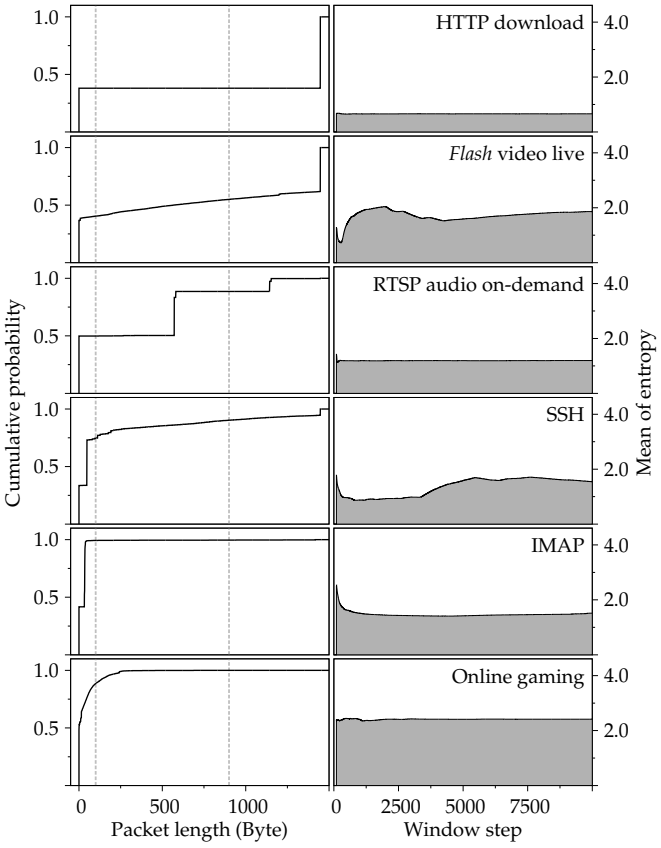

Fig. 2. Cumulative probability distribution of the packet lengths for examples of non-P2P flows and the corresponding mean of the entropy for a sliding window with size of 100 packets.

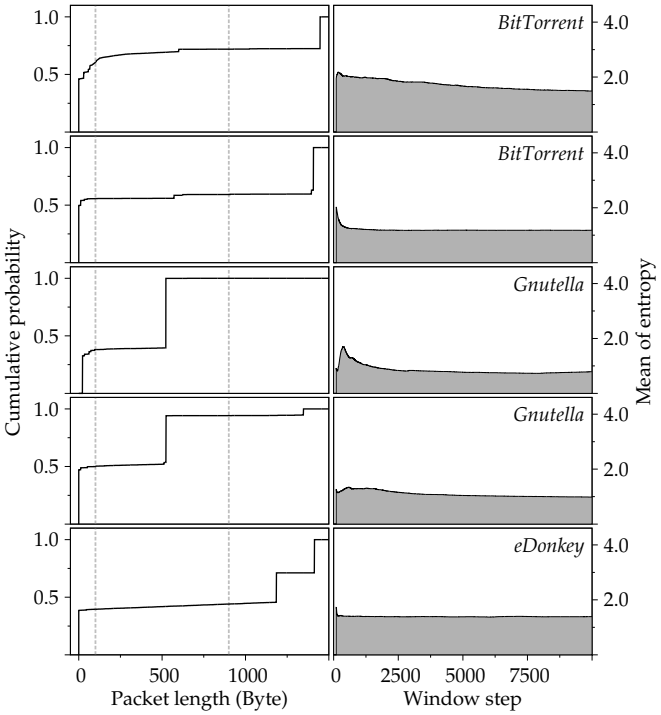

Fig. 3. Cumulative probability distribution of the packet lengths for examples of P2P file-sharing flows and the corresponding mean of the entropy for a sliding window with size of 100 packets.

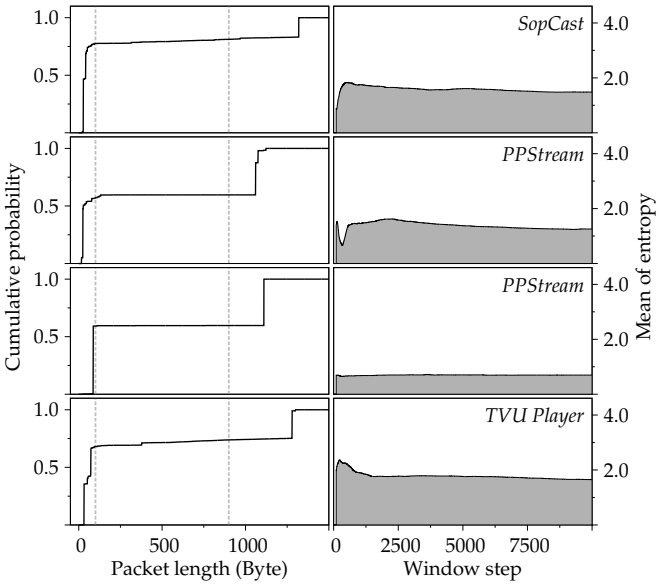

Fig. 4. Cumulative probability distribution of the packet lengths for examples of P2P video streaming and the corresponding mean of the entropy for a sliding window with size of 100 packets.

first 10000 packets and the mean of the entropy in each window iteration.

Typical *client-server* services, like the HTTP download of a large file, generate only a single or a few flows with very homogeneous packet lengths. However, other services like multimedia streaming also generated packets with different lengths besides of the two most frequent ones, as depicted in the second and third rows of Fig. 2. In other types of non-P2P applications, e.g., SSH, *Telnet*, or online games, there is a strong human behavior influence. The user commands or game instructions and the corresponding answers from the server generate several small packets with distinct lengths. Furthermore, mail traffic presents a typical *client-server* behavior when it results from the download of messages using POP or IMAP or from the upload of messages to be sent using SMTP. Each time a client checks for new messages in a POP server, a new flow is created. If there are no new messages (or only a very small number of small messages) in the server, the created flow will be too short to fill the sliding window. However, the synchronization between a client and an IMAP server keeps an active flow during the period the client application is running. In this case, the flow is formed by small packets with different lengths used, for example, to check the *inbox*, to delete messages, or to mark them as read.

Although the aggregation of P2P flows results in very heterogeneous traffic in terms of packet lengths, indi-

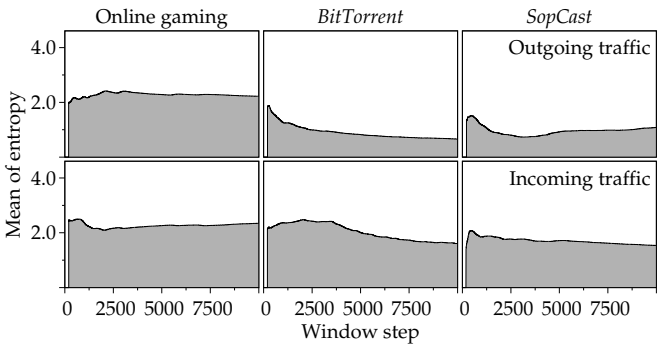

Fig. 5. Mean of the entropy for incoming and outgoing traffic for three application examples, using a sliding window with size of 100 packets.

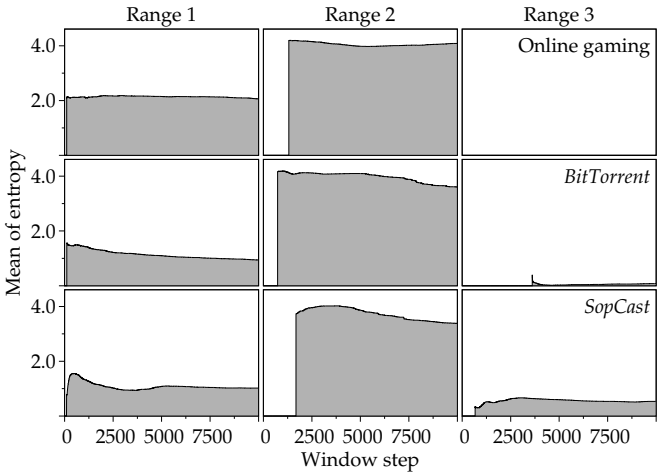

Fig. 6. Mean of the entropy, for all traffic, in three ranges of packet lengths for three examples, using a sliding window with size of 100 packets.

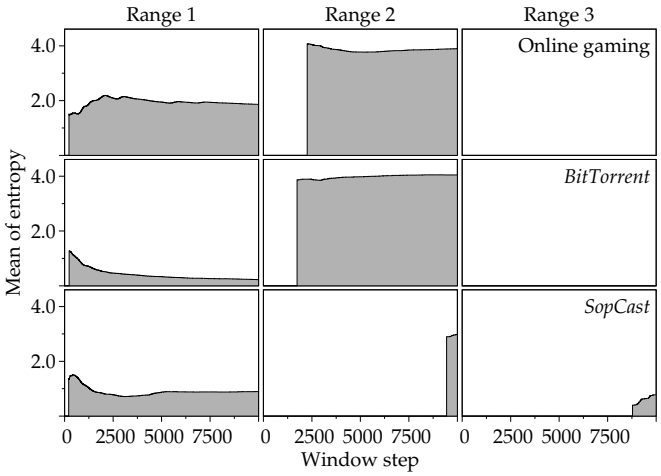

Fig. 7. Mean of the entropy, for outgoing traffic, in three ranges of packet lengths for three examples, using a sliding window with size of 100 packets.

vidually they have, in most cases, an entropy level very similar to the ones from non-P2P flows. Therefore, in order to identify differences between the P2P and non-P2P traffic, we studied the packet lengths in different dimensions of the traffic. We started by analyzing the entropy of the packet lengths separately (using separate sliding windows) for incoming and outgoing traffic. This approach allows us to explore the properties of the traffic in both ways, which may be distinct for both traffic classes as P2P applications are also used to provide contents. Additionally, it helps to characterize the packet lengths from non-P2P applications with strong human influence. Fig. 5 depicts the mean of the entropy for incoming and outgoing traffic for examples of non-P2P, P2P file-sharing, and P2P video streaming traffic in the first, second and third columns, respectively. Additional examples are included in Fig. 6 of the supplemental material. In typical *client-server* traffic, like the one generated by an HTTP download, almost all packet lengths in each direction are equal, which results in a low entropy in both directions, as depicted in Fig. 6 of the supplemental material. However, the randomness of the human actions of a game player increases the entropy of the outgoing traffic from online gaming. The P2P file-sharing and P2P video streaming present high entropy in both directions, being higher for incoming data though, mostly due to the contents requests and other small messages. The x axes in Fig. 5 represent the total of the steps of the two windows (incoming and outgoing traffic), which corresponds to the first 10000 packets. For each packet length, we update the sliding window of the corresponding direction (incoming or outgoing) and the resulting entropy, while the entropy for the opposite direction is kept intact.

The traffic flows from most applications, even the ones based on the P2P paradigm, contain a few more frequent packet lengths, which, in most cases, correspond to small, large, or medium-sized packets. Therefore, we also analyzed the entropy of the lengths of the packets in both directions separately for three different ranges of lengths: range 1 goes from 0 to 100 bytes, range 2 goes from 101 to 900 bytes, and range 3 goes from 901 to 1500 bytes. These ranges are marked with a gray dashed line in Figs. 2, 3, and 4, and, in the supplement material, in Figs. 1, 2, and 3. Furthermore, we also analyzed the entropy of the three ranges of packet lengths for only outgoing traffic. Figs. 6 and 7 present three examples of these analyses. Additional examples are depicted in Figs. 4 and 5 of the supplemental material. This approach enables the evaluation of the heterogeneity of the packet lengths in specific ranges, which give us more features to identify patterns to distinguish P2P traffic.

Using the analysis described in this section, we were able to create a set of rules to identify P2P traffic based on the entropy of the different dimensions of the packet lengths. For example, one of the rules is defined as:

> *if the mean of entropy for outgoing traffic is greater than 0.5 and the mean of entropy for the packets in*

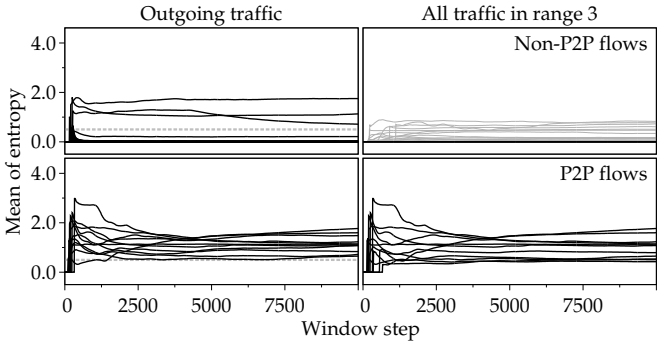

Fig. 8. Example of a rule for the identification of P2P traffic which uses the mean of the entropy of the packet lengths of outgoing and range 3 traffic.

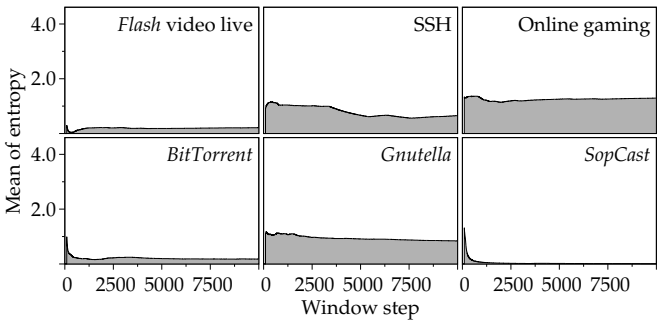

Fig. 9. Mean of the entropy of the inter-arrival times with a precision of 0.1 seconds.

range 3 is greater than 0, then the flow is P2P.

Fig. 8 illustrates this rule. The non-P2P flows represented in gray in the plots for the range 3 traffic correspond to the ones whose mean of entropy for outgoing traffic is below 0.5. The flows, with mean of entropy for outgoing traffic greater than 0.5, have the mean of entropy for range 3 packet lengths equal to zero. This rule alone was able to correctly identify between 47% and 50% of the P2P traffic in the datasets we described in section 3.1. The rules we defined are further described and discussed in section 4.

3.5 Entropy Analysis for Additional Features

The analysis of the packet lengths described in the previous subsection allowed us to define a set of rules based only on the entropy of the packet lengths. Using these rules we successfully identified between 95% and 98% of the P2P traffic included in the datasets described in section 3.1. However, the number of false positives was high, which dropped the total accuracy to values between 65% and 70%. Therefore, in order to improve the accuracy of the rules, we extended the analysis of the heterogeneity to additional traffic features. One of those features is the inter-arrival times that we analyzed with a precision of 0.1 seconds and computed the corresponding mean of the entropy. As a result of that analysis, Fig. 9 depicts the mean of the entropy for examples of

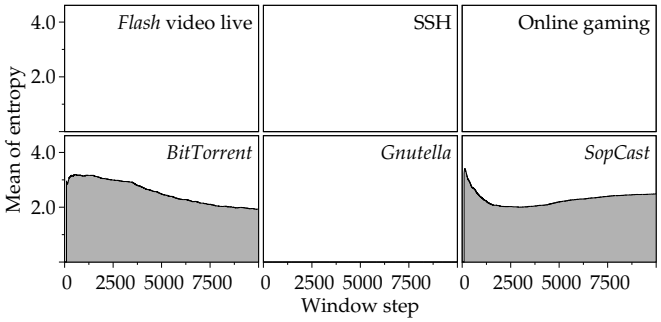

Fig. 10. Mean of the entropy of remote *host/port* pairs.

P2P and non-P2P traffic. The use of the mean of the entropy of the inter-arrival times to improve the rule performance led to a decrease of the P2P traffic identified in the test datasets but increased the total accuracy to more than 90%.

Additionally, since we analyzed the traffic separately for *host/port* pairs, we also computed the mean of the entropy of the remote *host/port* pairs that each pair communicates with. For each packet sent or received by a *host/port*, we identify the remote *host/port* and we evaluate how varied the remote *host/port* pairs are, by computing the entropy. This approach allows us to identify P2P *host/port* pairs that are providing contents to several peers at the same time. Fig. 10 represents the mean of the entropy for some examples of traffic. In the examples of *BitTorrent* and *SopCast host/port* pairs represented in this figure, the applications are exchanging data with several peers, whereas in the other examples, the applications are communicating with only a single remote *host, pair* and thus the entropy is zero. Using a rule based on this feature improved the accuracy of the identification of P2P traffic in the datasets.

4 P2P Traffic Classifier

The entropy analysis described in the previous section was used to implement a P2P traffic classifier. This section explains the operation of the classifier and describes the rules used by the classifier to distinguish between P2P and non-P2P traffic.

4.1 Classification Rules

The rules we have defined for the classifier use the following features:

- mean of the entropy of the packet lengths for incoming and outgoing traffic (ALL);
- mean of the entropy of the packet lengths for incoming and outgoing traffic in range 1 (ALLR1) and range 3 (ALLR3);
- mean of the entropy of the packet lengths for outgoing traffic (OUT);
- mean of the entropy of the packet lengths for outgoing traffic in range 1 (OUTR1);

TABLE 1
List of rules used by the classifier.

Number	Description	Class
1	IN > 1.5 and OUT > 0.5 and (OUT – OUTR1) < 0.01	Non-P2P
2	OUT > 2.5 and IN > 2.5	P2P
3	OUT > 0.5 and ALLR3 > 0	P2P
4	HP > 0.5	P2P
5	OUT = OUTR1 and ALLR3 > 0 and IAT < 0.5	Non-P2P
6	ALLR3 > 1	P2P
7	OUT > 1 and ALLR3 > 0	P2P
8	IAT > 1 and ALL < 1.5	P2P
9	ALL > 1 and ALLR1 = 0 and IAT > 0	P2P
10	ALLR1 < 0.5 and ALLR3 > 0.5 and IN < 2 and IAT > 0.1	P2P
11	ALL > 1 and OUT < 0.5 and IN < 1	
11a	if ALLR1 < 0.1 and ALLR3 < 1 and ((OUTR1 * 2) >= OUT or OUTR1 = 0	Non-P2P
	else	P2P
12	OUT > 0.001 and ALL < 1.5 and IAT > 0.1	P2P

- mean of the entropy of the packet lengths for incoming traffic (IN);
- mean of the entropy of the inter-arrival times with a precision of 0.1 seconds (IAT);
- mean of the entropy of remote *host/port* pairs (HP).

Table 1 contains a list of the rules used by the classifier. The classifier checks the rules sequentially from 1 to 12. If the traffic features match one of the rules, the traffic is identified as belonging to the corresponding class. Otherwise, the next rule is checked. If no rule is matched, the traffic is classified as non-P2P. To illustrate this process, we included the flowchart of the rules verification in appendix C of the supplemental material.

Most rules are simple and use only two or three features. In rule 1, we used the expression (OUT – OUTR1) < 0.01 to consider only the cases where OUT and OUTR1 are very close. This rule is useful for flows in which almost all outgoing packets belong to range 1. However, occasional packets from other ranges can make OUT slightly distinct from OUTR1. Hence, we used this expression to consider those cases.

Rule 2 was included to identify the traffic from P2P VoIP sessions. In [33], we defined a more complex strategy to deal with the traffic that results from the different speech codecs. Nonetheless, herein we defined only one rule to identify the traffic from common P2P VoIP applications, like *Skype* or *Google Talk*, since we are not focused on the identification of traffic from VoIP sessions with different speech codecs.

Rule 11 is more complex and can be better understood in the flowchart of the rules verification from Fig. 7 of the supplemental material. The first part is formed by the expression ALL > 1 and OUT < 0.5 and IN < 1. If this expression is matched, rule 11a is checked. The traffic is classified as non-P2P if rule 11a is matched. Otherwise, it is classified as P2P. If the first part is not matched, the next rule is checked.

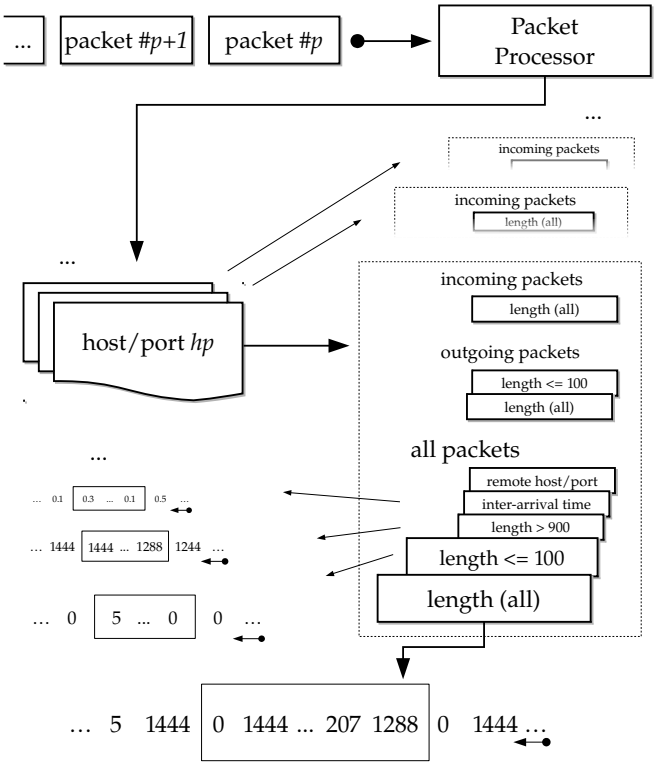

Fig. 11. Traffic analysis for each *host/port* pair, using independent sliding windows for the traffic features for which the entropy is evaluated.

During the definition of the rules, we gave priority to the traffic from some services or applications. For example, SSH sessions generate a small amount of data when compared to P2P file-sharing or video streaming. Some draft rules we tested were capable of accurately identifying a significant amount of P2P. However, they also classified SSH traffic as P2P. Depending on the classification purpose, it may be advantageous, in terms of the percentage of P2P traffic identified, to use rules that classify a large amount of P2P traffic, even if non-P2P flows with small packets are misclassified. Nonetheless, we consider important to give priority to non-P2P services like SSH, *Telnet*, or FTP.

Rule 4 is mostly used to identify P2P *host/port* pairs acting as servers. For this reason, every *host/port* running a server of a non-P2P service (e.g., HTTP, FTP, etc.) will also match this rule. Nonetheless, hosts running servers are usually well known nodes in monitored networks.

4.2 Structure and Operation of the Classifier

The operation of the proposed classifier is simple. It resorts to the entropy analysis explained in section 3 and it uses the rules described in the previous subsection. The traffic is processed, separated, and analyzed based on the *host/port* pair belonging to the monitored network. For each distinct *host/port*, the classifier keeps eight individual sliding windows, which are updated every time

a new packet sent or received by the *host/port* pairs is processed, as illustrated in Fig 11. The eight windows correspond to the features used by the rules in Table 1.

Therefore, the classifier maintains an updated record of the eight values of the mean of the entropy, for every processed *host/port*. The entropy is updated in real-time using the efficient method explained in section 3.2. For each analyzed packet, after updating the sliding windows and the corresponding entropies, the classifier tries to match a rule with the entropy levels and updates the classification of the *host/port* accordingly.

This approach makes possible to generate an updated classification result for every processed packet. In our study, we were primarily focused on the effectiveness of the classification and on the analysis of the traffic. Hence, we run the classifier with no constraints, outputting a log with the continuously updated classification results for every packet of all *host/port* pairs. However, the classifier can be set up to produce classifications for every packet only for the first n packets, or to output a classification only every n packets. It is expected that such strategy would improve the efficiency of the classification. Furthermore, the implementation includes two parameters that enable the periodical removing of the sliding windows from inactive *host/port* pairs, which decreases the consumption of the computing resources.

5 Performance Evaluation

In order to evaluate the performance of the classifier, we collected datasets containing traffic from several applications. We processed the datasets with the classifier and we compared the accuracy of the results against the ones obtained with other available tools. The following subsections describe the datasets used to evaluate the performance and analyze the accuracy of the classifier and its computational efficiency.

5.1 Datasets Used for Performance Evaluation

The evaluation of the accuracy of a traffic classifier raises a few problems. In order to verify if the obtained results are accurate, it is necessary to have previous knowledge of the applications that generated each flow. In most works, the ground truth information is obtained using a different classifier, usually a DPI-based one. This approach was not useful in our case, since the lack of effectiveness of the payload-based mechanisms when dealing with encrypted data is well known. Moreover, to effectively use a DPI method to obtain the datasets ground truth, it is necessary that the datasets contain the payload data. However, the availability of datasets with payload is scarce and capturing payload data in large networks is usually constrained by legal issues.

Therefore, we set up a testbed with several hosts and an aggregation point where the traffic generated by all of them was captured. The hosts were running *Microsoft Windows* or *Linux* operating systems and were connected in a LAN environment. We kept a record of the applications run by each host in each moment, so that we could use this information to evaluate the performance of the classifier. This approach allowed us to know exactly which applications generated each flow and avoid resorting to the accuracy of an external classifier. Moreover, we were also able to capture the payload information with no legal constraints, which is useful to compare the performance of the proposed classifier with the one achieved with a DPI method.

TABLE 2
Evaluation datasets.

Datasets	Volume (GB)	TCP (%)	UDP (%)
Dataset 1	8.80	78.35	21.59
Dataset 2	8.60	82.77	17.18
Dataset 3	9.97	77.13	22.84

Using the testbed, we captured 27.37 GB of traffic in different periods in August 2011, divided in three datasets as described in Table 2. The datasets contain traffic from all the applications listed in section 3.1, using, when available, protocol obfuscation or similar techniques. The composition of the datasets is further described in Table 2 of the supplemental material.

5.2 Performance of the Classifier

In order to evaluate the performance of the classifier, we analyzed the accuracy of the classification of the flows. Since the proposed classifier was designed to produce a classification for each window step, it was necessary to obtain a classification per flow. Hence, we considered as P2P the flows that were classified as P2P in at least 60% of the window steps. If the flow was classified as P2P in less than 40% of the window steps, we consider the flow to be non-P2P traffic. We considered the percentage between 40% and 60% as a *gray area*. Every flow classified as P2P in the range from 40% to 60% of the window steps was considered misclassified, independently of the true nature of the flow.

The performance of the classifier was evaluated using three metrics defined in [96], [97] and in the supplemental material: precision, recall, and accuracy. Precision evaluates how many of the cases classified as P2P were in fact P2P, recall evaluates how many of the true P2P cases were correctly identified by the classifier, and accuracy measures the overall performance of the classifier.

Many P2P applications generate a large number of very short flows with a small number of packets. For this reason, evaluating the performance simply by counting the number of correctly identified flows may not be representative of the capacity of a classifier. Correctly classifying a long flow with large packets is, in most cases, more interesting than classifying a few short flows with small packets. Additionally, some applications only generate flows with small packets (e.g., online gaming), while others also generate many large packets (e.g., HTTP download). Therefore, we identified the correctly

classified flows, counted the number of packets and the total amount of bytes transported in the flow, and evaluated the performance in terms of packets and bytes.

Table 3 summarizes the results of the performance evaluation. The classifier performs very well, with the precision and the accuracy above 95% in terms of bytes and almost 92% in terms of packets. Since the proposed mechanism resorts to sliding windows, it can only classify flows that have at least 100 packets (the size of the sliding window). For the sake of a rigorous analysis, we also included in Table 3 the results obtained if we exclude the flows with less than 100 packets.

The values of the recall for the different types of P2P applications were included in Table 4. As it is possible to see, almost all P2P video streaming was correctly classified. Most P2P VoIP traffic was also correctly classified, regardless of *Skype* being known for encrypting the traffic. Although the recall for P2P file-sharing is lower, it is still around 90%. Additionally, as explained before, we consider every case that falls in the *gray area* and every *host/port* pair with less than 100 packets as misclassified, regardless if it is P2P or non-P2P traffic. However, the results included in Table 4 were obtained only for the P2P traffic in datasets and, therefore, they are slightly different from the ones in Table 3.

After evaluating the performance of the proposed classifier, we compared it with other available classifiers. Besides of payload based classifiers, there are not many available and ready to use classifiers relying on behavioral methods. Recently, Lee et al. released *NeTraMark* [98], a framework that integrates several traffic classifiers. We used a few classifiers included in *NeTraMark* to classify the datasets. Table 5 summarizes the results obtained with *Blinc*, *C4.5* decision tree, SVMs, and Naïve Bayes. In Table 3 of the supplemental material, we included the results obtained with additional classifiers included in *NeTraMark*: port and payload-based, bayesian networks, and neural networks. The accuracy of most classifiers is around 50%. Some of them, like the port-based classifier, have a high precision, meaning that every flow classified as P2P was in fact P2P. However, the low recall shows that only a small percentage of the P2P traffic was correctly classified. In terms of recall, *Blinc* and Naïve Bayes achieved the best performance among the eight classifiers we tested.

5.3 Computational Efficiency

In order to evaluate how the consumption of computational resources scales as the traffic grows and indicate the processing and memory requirements of the proposed classifier, we followed the reasoning described below. The classification mechanism described herein separates the traffic based on host IP and port and to each *host/port* pair associates eight sliding windows. After the sliding windows are filled, no more information is saved in the memory. The oldest values leave the windows and new ones are added. Therefore, the memory requirements are directly proportional to the number of *host/port* pairs. Since the sliding windows from inactive *host/port* pairs are periodically removed, the memory used is directly proportional to the number of *host/port* pairs only in the worst case scenario in which all *host/port* pairs are active.

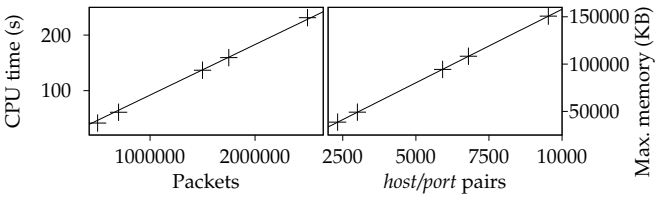

Fig. 12. Representation of the CPU time and maximum memory consumption growing regarding the number of packets and distinct *host/port* pairs.

Every time a new packet arrives, it is processed by the classifier, the sliding windows are updated, and new entropy values are generated. This process is repeated for every packet, regardless of the *host/port* pairs. Hence, the processing power required by the tool is directly proportional to the number of packets analyzed.

Although the implementation of the proposed classifier is not optimized, we made several measurements of the CPU and memory consumption during execution of the classifier when processing trace files containing different amounts of packets and *host/port* pairs, using `/usr/bin/time` tool. This analysis gives us information about how the consumption of resources grows when the traffic increases.

We extracted five trace files with different sizes from the datasets previously used to evaluate the classifier performance. Table 4 of the supplemental material presents the number of packets and distinct *host/port* pairs of each of the traces files and the CPU time and memory measurements used by the classifier. In order to evaluate the worst case scenario, we modified the classifier to not remove the sliding windows of the inactive *host/port* pairs. The linear dependency of the CPU time and the number of packets, and of the memory used and number of distinct *host/port* pairs is observable in Fig. 12.

6 Conclusion

In this article, we proposed a new mechanism for P2P traffic classification that is mostly based on the heterogeneity of the packet lengths. Unlike our previous works, this classifier is able to identify P2P flows instead of only hosts running P2P applications. The method analyzes the packet lengths in three different ranges and for both flow directions. The entropy is used as a measure of the heterogeneity of the analyzed features and it is computed using a sliding window with a constant size of 100 packets. This approach allows the classifier to obtain a result for each packet, making it suitable for real-time traffic classification. Since it does not use any payload

TABLE 3
Results of the performance evaluation of the proposed classifier.

Datasets	All *host/port* pairs						Excluding *host/port* pairs with less than 100 packets					
	Bytes			Packets			Bytes			Packets		
	Accur.	Precision	Recall	Accur.	Precision	Recall	Accur.	Precision	Recall	Accur.	Precision	Recall
Dataset 1	95.68%	95.87%	92.53%	92.54%	91.96%	89.46%	96.72%	97.91%	94.43%	96.63%	98.27%	95.41%
Dataset 2	96.44%	95.27%	93.22%	93.40%	91.99%	90.07%	97.62%	97.93%	95.77%	97.00%	98.08%	95.91%
Dataset 3	96.49%	96.21%	93.88%	93.88%	93.02%	91.08%	97.30%	97.86%	95.46%	97.07%	98.18%	96.02%

TABLE 4
Recall results for P2P traffic, in terms of bytes.

Datasets	P2P	P2P File-Sharing	P2P Streaming	P2P VoIP
Dataset 1	92.90%	88.03%	99.87%	94.68%
Dataset 2	94.50%	91.53%	99.81%	98.27%
Dataset 3	94.34%	89.99%	99.90%	98.60%

data, the mechanism is effective with encrypted traffic. Based on the entropy analysis, we defined a set of rules used by the classifier to identify P2P traffic. In order to improve the accuracy in specific cases, the mechanism also measures the entropy for inter-arrival times and remote *host/port* pairs.

The performance of the classifier was evaluated using a few datasets collected in a testbed. The ground truth information was saved so that it could be possible to know exactly which application generated each flow. The results show that the classifier was able to identify the P2P traffic with very high precision. The recall rate demonstrates that almost all P2P data in the datasets was correctly classified. Additionally, we included an evaluation of the classifier efficiency that helps to understand how the resources consumption grows when the traffic increases. The CPU time and used memory increase linearly with the amount analyzed data.

Acknowledgments

This work was partially supported by University of Beira Interior, by *Instituto de Telecomunicações*, and by the portuguese *Fundação para a Ciência e a Tecnologia*, through the grant contract SFRH/BD/60654/2009.

References

[1] A. Kind, X. Dimitropoulos, S. Denazis, and B. Claise, "Advanced network monitoring brings life to the awareness plane," *IEEE Commun. Mag.*, vol. 46, no. 10, pp. 140–146, Oct. 2008.

[2] G. Goth, "Traffic management becoming high-priority problem," *IEEE Internet Comput.*, vol. 12, no. 6, pp. 6–8, Nov./Dec. 2008.

[3] M. Mellia, A. Pescapè, and L. Salgarelli, "Traffic classification and its applications to modern networks," *Elsevier Comput. Netw.*, vol. 53, no. 6, pp. 759–760, Apr. 2009.

[4] B. Krishnamurthy and J. Wang, "Traffic classification for application specific peering," in *Proc. ACM SIGCOMM Internet Measurement Workshop (IMW 2002)*, Marseille, France, Nov. 2002, pp. 179–180.

[5] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proc. 16th Annual Multimedia Computing and Networking (MMCN '02)*, San Jose, CA, USA, Jan. 2002.

[6] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, Apr. 2004.

[7] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Measurement*, ser. LNCS. Springer Berlin / Heidelberg, 2005, vol. 3431, pp. 41–54.

[8] A. Callado, C. Kamienski, G. Szabó, B. P. Gero, J. Kelner, F. Stênio, and D. Sadok, "A survey on Internet traffic identification," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 37–52, Jul.–Sep. 2009.

[9] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus, "Lightweight, payload-based traffic classification: An experimental evaluation," in *Proc. IEEE Int. Conf. Communications (ICC 2008)*, Beijing, China, May 2008, pp. 5869–5875.

[10] R. Smith, C. Estan, S. Jha, and S. Kong, "Deflating the big bang: Fast and scalable deep packet inspection with extended finite automata," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 207–218, Oct. 2008.

[11] A. Esteves, P. R. M. Inácio, M. Pereira, and M. M. Freire, "On-line detection of encrypted traffic generated by mesh-based peer-to-peer live streaming applications: The case of GoalBit," in *Proc. 10th IEEE Int. Symp. Network Computing and Applications (NCA 2011)*, Cambridge, MA, USA, Aug. 2011, pp. 223–228.

[12] P. Ohm, D. C. Sicker, and D. Grunwald, "Legal issues surrounding monitoring during network research," in *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC 2007)*, San Diego, CA, USA, Oct. 2007, pp. 141–148.

[13] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Passive and Active Network Measurement*, ser. LNCS. Springer Berlin / Heidelberg, 2007, vol. 4427, pp. 165–175.

[14] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting," *Elsevier Comput. Netw.*, vol. 53, no. 1, pp. 81–97, Jan. 2009.

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

[16] W. H. Turkett, A. V. Karode, and E. W. Fulp, "In-the-dark network traffic classification using support vector machines," in *Proc. 20th Conf. Innovative Applications of Artificial Intelligence (IAAI '08)*, Chicago, IL, USA, Jul. 2008, pp. 1745–1750.

[17] F. Constantinou and P. Mavrommatis, "Identifying known and unknown peer-to-peer traffic," in *Proc. 5th IEEE Int. Symp. Network Computing and Applications (NCA 2006)*, Cambridge, MA, USA, Jul. 2006, pp. 93–102.

[18] W. John and S. Tafvelin, "Heuristics to classify Internet backbone traffic based on connection patterns," in *Proc. Int. Conf. Information Networking (ICOIN 2008)*, Busan, Korea, Jan. 2008, pp. 1–5.

[19] J. Cao, A. Chen, I. Widjaja, and N. Zhou, "Online identification of applications using statistical behavior analysis," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2008)*, New Orleans, LA, USA, Nov./Dec. 2008, pp. 1–6.

[20] A. Bianco, G. Mardente, M. Mellia, M. Munafò, and L. Muscariello, "Web user-session inference by means of clustering techniques," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 405–416, Apr. 2009.

[21] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Elsevier Comput. Netw.*, vol. 53, no. 14, pp. 2476–2490, Sep. 2009.

[22] N. Cascarano, A. Este, F. Gringoli, F. Risso, and L. Salgarelli, "An experimental evaluation of the computational cost of a DPI traffic classifier," in *Proc. IEEE Global Communications Conf. (GLOBECOM 2009)*, Honolulu, HI, USA, Nov./Dec. 2009.

TABLE 5
Results of the performance evaluation of other classifiers, in terms of bytes.

Datasets	*Blinc*			*C4.5* Decision Tree			Support Vector Machines			Naïve Bayes		
	Accur.	Precision	Recall	Accur.	Precision	Recall	Accur.	Precision	Recall	Accur.	Precision	Recall
Dataset 1	55.60%	59.80%	50.25%	44.84%	41.77%	09.61%	46.43%	43.79%	02.78%	47.01%	50.10%	75.78%
Dataset 2	51.92%	46.40%	47.08%	50.83%	33.46%	09.86%	50.04%	47.48%	03.33%	51.17%	47.21%	76.19%
Dataset 3	61.02%	64.35%	48.48%	49.93%	47.87%	07.35%	50.60%	61.49%	01.85%	49.60%	49.57%	75.89%

[23] M. E. Johnson, D. McGuire, and N. D. Willey, "The evolution of the peer-to-peer file sharing industry and the security risks for users," in *Proc. 41st Hawaii International Conf. System Sciences (HICSS 2008)*, Waikoloa, HI, USA, Jan. 2008, pp. 1–10.

[24] D. Chopra, H. Schulzrinne, E. Marocco, and E. Ivov, "Peer-to-peer overlays for real-time communication: Security issues and solutions," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 4–12, Jan.–Mar. 2009.

[25] Z. Chen, B. Yang, Y. Chen, A. Abraham, C. Grosan, and L. Peng, "Online hybrid traffic classifier for peer-to-peer systems based on network processors," *Elsevier Appl. Soft Comput.*, vol. 9, no. 2, pp. 685–694, Mar. 2009.

[26] Y. Zhang, H. Wang, and S. Cheng, "A method for real-time peer-to-peer traffic classification based on C4.5," in *Proc. 12th IEEE Int. Conf. Communication Technology (ICCT 2010)*, Ibaraki, Japan, Nov. 2010, pp. 1192–1195.

[27] C. Gu and S. Zhuang, "A novel P2P traffic classification approach using back propagation neural network," in *Proc. 12th IEEE Int. Conf. Communication Technology (ICCT 2010)*, Nanjing, China, Nov. 2010, pp. 52–55.

[28] K. Xu, M. Zhang, M. Ye, D. M. Chiu, and J. Wu, "Identify P2P traffic by inspecting data transfer behavior," *Elsevier Comput. Commun.*, vol. 33, no. 10, pp. 1141–1150, Jun. 2010.

[29] R. Keralapura, A. Nucci, and C.-N. Chuah, "A novel self-learning architecture for p2p traffic classification in high speed networks," *Elsevier Comput. Netw.*, vol. 54, no. 7, pp. 1055–1068, May 2010.

[30] J. Hurley, E. Garcia-Palacios, and S. Sezer, "Host-based P2P flow identification and use in real-time," *ACM Trans. Web*, vol. 5, pp. 1–27, May 2011.

[31] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti, "Abacus: Accurate behavioral classification of P2P-TV traffic," *Elsevier Comput. Netw.*, vol. 55, no. 6, pp. 1394–1411, Apr. 2011.

[32] J. V. P. Gomes, P. R. M. Inácio, M. M. Freire, M. Pereira, and P. P. Monteiro, "Analysis of peer-to-peer traffic using a behavioural method based on entropy," in *Proc. 27th IEEE Int. Performance Computing and Communications Conf. (IPCCC 2008)*, Austin, TX, USA, Dec. 2008, pp. 201–208.

[33] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, "Identification of peer-to-peer VoIP sessions using entropy and codec properties," *submitted for publication*.

[34] ——, "Exploring behavioral patterns through entropy in multimedia peer-to-peer traffic," *The Computer Journal*, vol. 55, no. 6, pp. 740–755, Jun. 2012.

[35] G. Aceto, A. Dainotti, W. de Donato, and A. Pescapé, "PortLoad: Taking the best of two worlds in traffic classification," in *Proc. 29th IEEE Conf. Computer Communications Workshops (INFOCOM 2010)*, San Diego, CA, USA, Mar. 2010, pp. 1–5.

[36] C. Gu, S. Zhang, and X. Xue, "Encrypted internet traffic classification method based on host behavior," *Int. J. Digital Content Technol. Appl.*, vol. 5, no. 3, pp. 167–174, Mar. 2011.

[37] H. Pereira, A. Ribeiro, and P. Carvalho, "Improving traffic classification and policing at application layer," in *Proc. IEEE Symp. Computers and Communications (ISCC 2010)*, Riccione, Italy, Jun. 2010, pp. 291–294.

[38] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in *Proc. IEEE/IFIP Network Operations and Management Symp. (NOMS 2008)*, Salvador da Bahia, Brazil, Apr. 2008, pp. 160–167.

[39] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "KISS: Stochastic packet inspection classifier for UDP traffic," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1505–1515, Oct. 2010.

[40] G. L. Mantia, D. Rossi, A. Finamore, M. Mellia, and M. Meo, "Stochastic packet inspection for TCP traffic," in *Proc. IEEE Int. Conf. Communications (ICC 2010)*, Cape Town, South Africa, May 2010, pp. 1–6.

[41] S. Kumar, S. Dharmapurikar, F. Yu, P. Crowley, and J. Turner, "Algorithms to accelerate multiple regular expressions matching for deep packet inspection," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 339–350, Oct. 2006.

[42] T. Liu, Y. Sun, and L. Guo, "Fast and memory-efficient traffic classification with deep packet inspection in CMP architecture," in *Proc. 5th IEEE Int. Conf. Networking, Architecture and Storage (NAS 2010)*, Macau, China, Jul. 2010, pp. 208–217.

[43] N. Cascarano, L. Ciminiera, and F. Risso, "Optimizing deep packet inspection for high-speed traffic analysis," *J. Netw. Syst. Manag.*, vol. 19, no. 1, pp. 7–31, Mar. 2011.

[44] B. Park, J. W.-K. Hong, and Y. J. Won, "Toward fine-grained traffic classification," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 104–111, Jul. 2011.

[45] P. Dorfinger, G. Panholzer, B. Trammell, and T. Pepe, "Entropy-based traffic filtering to support real-time Skype detection," in *Proc. 6th Int. Wireless Communications and Mobile Computing Conf. (IWCMC '10)*, Caen, France, Jun./Jul. 2010, pp. 747–751.

[46] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: When randomness plays with you," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, Oct. 2007.

[47] G. Bartlett, J. Heidemann, and C. Papadopoulos, "Inherent behaviors for on-line detection of peer-to-peer file sharing," in *Proc. IEEE Global Internet Symp.*, Anchorage, AK, USA, May 2007, pp. 55–60.

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

[49] C.-C. Wu, K.-T. Chen, Y.-C. Chang, and C.-L. Lei, "Peer-to-peer application recognition based on signaling activity," in *Proc. IEEE Int. Conf. Communications (ICC 2009)*, Dresden, Germany, Jun. 2009, pp. 1–5.

[50] W. Cheng, J. Gong, W. Ding, and Z. Sun, "Application type identification of internet flows based on medium mathematics," *J. China Universities Posts Telecommun.*, vol. 17, no. 6, pp. 72–79, Dec. 2010.

[51] E. P. Freire, A. Ziviani, and R. M. Salles, "Detecting VoIP calls hidden in web traffic," *IEEE Trans. Netw. Service Manag.*, vol. 5, no. 4, pp. 204–214, Dec. 2008.

[52] F. Palmieri and U. Fiore, "A nonlinear, recurrence-based approach to traffic classification," *Elsevier Comput. Netw.*, vol. 53, no. 6, pp. 761–773, Apr. 2009.

[53] Y.-D. Lin, C.-N. Lu, Y.-C. Lai, W.-H. Peng, and P.-C. Lin, "Application classification using packet size distribution and port association," *J. Netw. Comput. Appl.*, vol. 32, no. 5, pp. 1023–1030, Sep. 2009.

[54] M. Dusi, A. Este, F. Gringoli, and L. Salgarelli, "Coarse classification of internet traffic aggregates," in *Proc. IEEE Int. Conf. Communications (ICC 2010)*, Cape Town, South Africa, May 2010, pp. 1–6.

[55] X. Wang and D. J. Parish, "Optimised multi-stage TCP traffic classifier based on packet size distributions," in *Proc. 3rd Int. Conf. Communication Theory, Reliability, and Quality of Service (CTRQ 2010)*, Athens, Greece, Jun. 2010, pp. 98–103.

[56] M. Iliofotou, H. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, "Graption: A graph-based P2P traffic classification framework for the internet backbone," *Elsevier Comput. Netw.*, vol. 55, no. 8, pp. 1909–1920, Jun. 2011.

[57] M. Hirvonen and J.-P. Laulajainen, "Two-phased network traffic classification method for quality of service management," in *Proc.*

13th IEEE Int. Symp. on Consumer Electronics (ISCE 2009), Kyoto, Japan, May 2009, pp. 962–966.

[58] Y. Chen, X. Ping, and T. Wei, "Application traffic classification based on command exchange mode of TCP flows," in *Proc. IEEE Int. Conf. Information Theory and Information Security (ICITIS 2010)*, Beijing, China, Dec. 2010, pp. 585–591.

[59] J. Hurley, E. Garcia-Palacios, and S. Sezer, "Classifying network protocols: A 'two-way' flow approach," *IET Commun.*, vol. 5, no. 1, pp. 79–89, Jan. 2011.

[60] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. ACM SIGCOMM Workshop Mining Network Data (MineNet '06)*, Pisa, Italy, Sep. 2006, pp. 281–286.

[61] T. T. T. Nguyen and G. Armitage, "Clustering to assist supervised machine learning for real-time IP traffic classification," in *Proc. IEEE Int. Conf. Communications (ICC '08)*, Beijing, China, May 2008, pp. 5857–5862.

[62] L. Bernaille, R. Teixeira, I. Akodjenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, Apr. 2006.

[63] G. Dewaele, Y. Himura, P. Borgnat, K. Fukuda, P. Abry, O. Michel, R. Fontugne, K. Cho, and H. Esaki, "Unsupervised host behavior classification from connection patterns," *Int. J. Netw. Manag.*, vol. 20, no. 5, pp. 317–337, Sep./Oct. 2010.

[64] Y. Lim, H. Kim, J. Jeong, C. Kim, T. T. Kwon, and Y. Choi, "Internet traffic classification demystified: On the sources of the discriminative power," in *Proc. 6th ACM SIGCOMM Int. Conf. emerging Networking EXperiments and Technologies (CoNEXT 2010)*, Philadelphia, PA, USA, Nov./Dec. 2010, pp. 1–12.

[65] B. Raahemi, W. Zhong, and J. Liu, "Peer-to-peer traffic identification by mining IP layer data streams using concept-adapting very fast decision tree," in *Proc. 20th IEEE Int. Conf. Tools with Artificial Intelligence (ICTAI 2008)*, vol. 1, Dayton, OH, USA, Nov. 2008, pp. 525–532.

[66] G. Mingliang, H. Xiaohong, T. Xu, M. Yan, and W. Zhenhua, "Data stream mining based real-time high-speed traffic classification," in *Proc. 2nd IEEE Int. Conf. Broadband Network Multimedia Technology (IC-BNMT 2009)*, Beijing, China, Oct. 2009, pp. 700–705.

[67] M. Sun, Y. Zhang, J. Chen, and T. Shi, "A P2P traffic identification method based on VFDT," in *Proc. Int. Conf. Intelligent Computation Technology and Automation (ICICTA 2010)*, vol. 1, Changsha, China, May 2010, pp. 281–284.

[68] Y.-H. Wang, V. Gau, T. Bosaw, J.-N. Hwang, A. Lippman, D. Liebennan, and I.-C. Wu, "Generalization performance analysis of flow-based peer-to-peer traffic identification," in *Proc. IEEE Workshop Machine Learning for Signal Processing (MLSP 2008)*, Cancún, Mexico, Oct. 2008, pp. 267–272.

[69] A. Dainotti, W. de Donato, A. Pescapè, and P. S. Rossi, "Classification of network traffic via packet-level hidden markov models," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2008)*, New Orleans, LA, USA, Nov./Dec. 2008, pp. 1–5.

[70] J. E. B. Maia and R. H. Filho, "Internet traffic classification using a hidden markov model," in *Proc. 10th Int. Conf. Hybrid Intelligent Systems (HIS 2010)*, Atlanta, GA, USA, Aug. 2010, pp. 37–42.

[71] M. Hong, R. Gu, H. Wang, Y. Sun, and Y. Ji, "Identifying online traffic based on property of TCP flow," *J. China Universities Posts Telecommun.*, vol. 16, no. 3, pp. 84–88, Jun. 2009.

[72] L. Huixian and L. Xiaojuan, "A novel traffic classification algorithm using machine learning," in *Proc. 2nd IEEE Int. Conf. Broadband Network Multimedia Technology (IC-BNMT 2009)*, Beijing, China, Oct. 2009, pp. 340–344.

[73] L. Peng, H. Zhang, B. Yang, Y. Chen, M. T. Qassrawi, and G. Lu, "Traffic identification using flexible neural trees," in *Proc. 18th IEEE Int. Workshop Quality of Service (IWQoS 2010)*, Beijing, China, Jun. 2010, pp. 1–5.

[74] S. Huang, K. Chen, C. Liu, A. Liang, and H. Guan, "A statistical-feature-based approach to internet traffic classification using machine learning," in *Proc. Int. Conf. Ultra Modern Telecommunications (ICUMT 2009)*, St. Petersburg, Russia, Oct. 2009, pp. 1–6.

[75] W. Jiang and M. Gokhale, "Real-time classification of multimedia traffic using FPGA," in *Proc. Int. Conf. Field Programmable Logic and Applications (FPL 2010)*, Milano, Italy, Aug./Sep. 2010, pp. 56–63.

[76] M. Kohara, Y. Hori, K. Sakurai, H. Lee, and J.-C. Ryou, "Flow traffic classification with support vector machine by using payload length," in *Proc. 2nd Int. Conf. Computer Science and its Applications (CSA 2009)*, Jeju Island, Korea, Dec. 2009, pp. 1–5.

[77] F. Liu, Z. Li, and Q. Nie, "A new method of P2P traffic identification based on support vector machine at the host level," in *Proc. Int. Conf. Information Technology and Computer Science (ITCS 2009)*, vol. 2, Kiev, Ukraine, Jul. 2009, pp. 579–582.

[78] C. H. Park and M. Lee, "A SVM-based discretization method with application to associative classification," *Elsevier Expert Syst. Appl.*, vol. 36, no. 3, pp. 4784–4787, Apr. 2009.

[79] R. Yuan, Z. Li, X. Guan, and L. Xu, "An SVM-based machine learning method for accurate internet traffic classification," *Inf. Syst. Frontiers*, vol. 12, no. 2, pp. 149–156, Apr. 2010.

[80] X. Li, F. Qi, D. Xu, and X. Qiu, "An internet traffic classification method based on semi-supervised support vector machine," in *Proc. IEEE Int. Conf. Communications (ICC 2011)*, Kyoto, Japan, Jun. 2011, pp. 1–6.

[81] J. Zhao, X. Huang, Q. Sun, and Y. Ma, "Real-time feature selection in traffic classification," *J. China Universities Posts Telecommun.*, vol. 15, no. 1, pp. 68–72, Sep. 2008.

[82] X. Wang and D. J. Parish, "Optimised TCP traffic classification with multiple statistical algorithms," in *Proc. Int. Conf. Information Networking and Automation (ICINA 2010)*, vol. 1, Kunming, China, Oct. 2010, pp. 261–265.

[83] J. Cai, Z. Zhang, and X. Song, "An analysis of UDP traffic classification," in *Proc. 12th IEEE Int. Conf. Communication Technology (ICCT 2010)*, Nanjing, China, Nov. 2010, pp. 116–119.

[84] A. Callado, J. Kelner, D. Sadok, C. A. Kamienski, and S. Fernandes, "Better network traffic identification through the independent combination of techniques," *Elsevier J. Netw. Comput. Appl.*, vol. 33, no. 4, pp. 433–446, Jul. 2010.

[85] R. Bar-Yanai, M. Langberg, D. Peleg, and L. Roditty, "Realtime classification for encrypted traffic," in *Proc. 9th Int. Symp. Experimental Algorithms (SEA 2010)*, ser. LNCS, vol. 6049, Napoli, Italy, May 2010, pp. 373–385.

[86] J. Li, S. Zhang, C. Li, and J. Yan, "Composite lightweight traffic classification system for network management," *Int. J. Netw. Manag.*, vol. 20, no. 2, pp. 85–105, Mar./Apr. 2010.

[87] 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 *Proc. 9th IEEE/ACIS Int. Conf. Computer and Information Science (ICIS 2010)*, Kaminoyama, Japan, Aug. 2010, pp. 105–110.

[88] M. Mohammadi, B. Raahemi, A. Akbari, H. Moeinzadeh, and B. Nasersharif, "Genetic-based minimum classification error mapping for accurate identifying peer-to-peer applications in the internet traffic," *Elsevier Expert Syst. Appl.*, vol. 38, no. 6, pp. 6417–6423, Jun. 2011.

[89] R. Alshammari and A. N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?" *Elsevier Comput. Netw.*, vol. 55, no. 6, pp. 1326–1350, Apr. 2011.

[90] W. Lu, M. Tavallaee, and A. A. Ghorbani, "Hybrid traffic classification approach based on decision tree," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM 2009)*, Honolulu, HI, USA, Nov./Dec. 2009, pp. 1–6.

[91] L. Jun, Z. Shunyi, L. Yanqing, and Y. Junrong, "Hybrid Internet traffic classification technique," *J. Electronics (China)*, vol. 26, no. 1, pp. 101–112, Jan. 2009.

[92] F. Rodríguez-Teja, C. Martinez-Cagnazzo, and E. G. Castro, "Bayesian classification: Methodology for network traffic classification combination," in *Proc. 6th Int. Wireless Communications and Mobile Computing Conf. (IWCMC 2010)*, Caen, France, Jun./Jul. 2010, pp. 769–773.

[93] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-time traffic classification based on statistical and payload content features," in *Proc. Int. Workshop Intelligent Systems and Applications (ISA 2010)*, Wuhan, China, May 2010, pp. 1–4.

[94] B. Li, M. Ma, and Z. Jin, "A VoIP traffic identification scheme based on host and flow behavior analysis," *J. Netw. Syst. Manag.*, vol. 19, no. 1, pp. 111–129, Mar. 2011.

[95] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical J.*, vol. 27, pp. 379–423, Jul. 1948.

[96] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, Feb. 1999, pp. 249–252.

[97] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*, 1st ed. Springer, Mar. 2008.

[98] S. Lee, H. Kim, D. Barman, S. Lee, C. Kim, T. T. Kwon, and Y. Choi, "NeTraMark: A network traffic classification benchmark," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 22–30, Jan. 2011.

A Supplement to "Classification of One-to-Many Peer-to-Peer Traffic Using Packet Length and Entropy"

João V. Gomes, Pedro R. M. Inácio, Manuela Pereira, Mário M. Freire, and Paulo P. Monteiro

Abstract—This supplement is organized as follows. Appendix A presents a more complete description of the composition of the datasets used to study the traffic properties. Appendix B includes additional examples of the traffic analysis described in the main article. Appendix C presents a flowchart of the process of rules matching used by the classifier. Appendix D describes the composition of the datasets and the metrics used in the performance evaluation of the proposed mechanism, presents the results obtained with four additional classifiers, and summarizes the evaluation of the consumption of computational resources.

Index Terms—Data communications, distributed applications, network communications, network management, network monitoring, packet-switching networks.

Appendix A
Experimental Datasets

The experimental datasets, collected in individual hosts and used to study the traffic properties, contain traffic from several distinct Internet applications or services, as explained in the main article. Table 1 describes the amount of traffic from each type of application. Peer-to-peer (P2P) traffic accounts for nearly 60% of the traffic in bytes and almost 70% in packets. The web browsing class includes only the traffic that results from browsing web pages. Other web contents, like video streaming using *Flash* technology, are included in the streaming class. The download of larges files using Hypertext Transfer Protocol (HTTP), like an executable or an disc image file, is also included in a separate class.

TABLE 1
Share of each type of traffic in the experimental data.

Traffic	Packets (%)			Bytes (%)		
		TCP	UDP		TCP	UDP
P2P file-sharing	28.30	26.28	01.94	28.53	27.41	01.11
P2P streaming	40.81	00.46	40.24	30.02	00.49	29.51
P2P VoIP	00.51	00.03	00.49	00.12	00.00	00.11
Mail	04.51	04.50	00.00	06.07	06.07	00.00
HTTP download	00.96	00.96	00.00	01.46	01.46	00.00
Web browsing	00.73	00.73	00.00	00.82	00.82	00.00
Streaming	21.62	21.41	00.20	30.47	30.36	00.10
Telnet / SSH	00.19	00.19	00.00	00.09	00.09	00.00
FTP / SFTP	01.50	01.50	00.00	02.21	02.21	00.00
Online gaming	00.87	00.87	00.00	00.21	00.21	00.00

Appendix B
Heterogeneity of Packet Lengths

The main article explains the properties of the packet lengths for different types of traffic and presents a few examples of the mean of the corresponding entropy. In this appendix, we include additional examples. Figs. 1, 2, and 3 depict the lengths of the packets and the corresponding entropy (instead of the mean) through all the window steps. The dashed lines in the figures mark the limits of three ranges of packet lengths: from 0 to 100 bytes, 101 to 900 bytes, and 901 to 1500 bytes. The representation makes it possible to see how entropy varies for different patterns of packets lengths. When the packet lengths are more heterogeneous, the entropy increases, as it is observable, for example, in the plot of the Secure Shell (SSH) session.

Figs. 4 and 5 present the mean of the entropy for different ranges of packet lengths for all traffic and outgoing traffic, respectively. Unless the flow is being used to provide contents, there are usually no packets from range 3 in outgoing direction. Since in an HTTP download there are only acknowledgments with the same length in outgoing direction, the mean of the entropy is zero also in range 1. Fig. 6 depicts the mean of the entropy for incoming and outgoing traffic. Since HTTP traffic is formed by packets with the same length in each direction, the mean of the entropy for the traffic in both directions is zero.

- *J. Gomes, P. Inácio, M. Pereira, and M. Freire are with Instituto de Telecomunicações, Department of Computer Science, University of Beira Interior, Portugal.*
  E-mail: jgomes@penhas.di.ubi.pt, {inacio, mpereira, mario}@di.ubi.pt
- *P. Monteiro is with Nokia Siemens Networks Portugal, S. A., with University of Aveiro, and with Instituto de Telecomunicações.*
  E-mail: paulo.1.monteiro@nsn.com

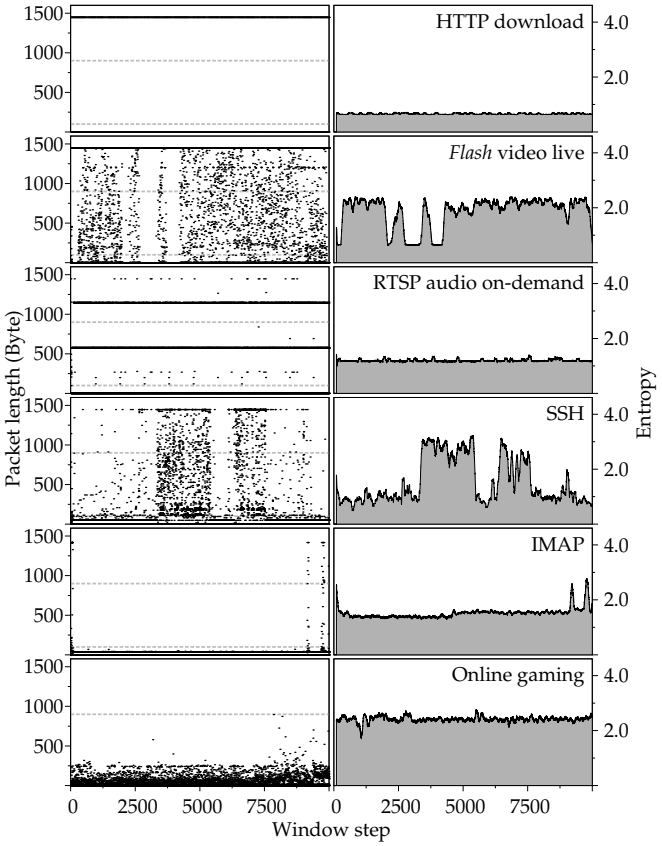

Fig. 1. Representation of the packet lengths for examples of non-P2P flows and the corresponding entropy for a sliding window with size of 100 packets.

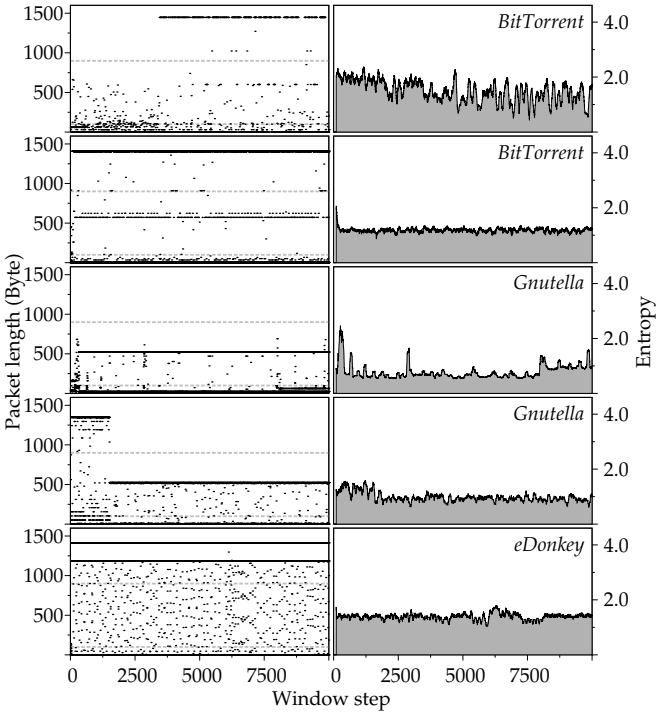

Fig. 2. Representation of the packet lengths for examples of P2P file-sharing flows and the corresponding entropy for a sliding window with size of 100 packets.

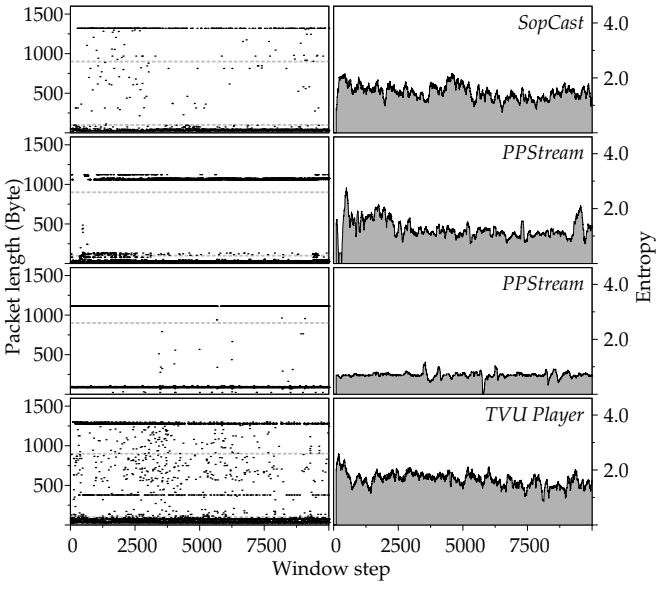

Fig. 3. Representation of the packet lengths for examples of P2P video streaming and the corresponding entropy for a sliding window with size of 100 packets.

Appendix C
Classifier Rules

The classifier described in the article uses a set of rules for P2P traffic identification. The rules are checked sequentially in ascending order, as represented in the flowchart of the rules matching process from Fig. 7. Each rule decision either results in a classification or proceeds to the next rule. The only exceptions are rule 11, which proceeds to rule 11a or rule 12, and rule 11a that results in a classification as P2P or non-P2P.

Appendix D
Performance Evaluation

In this appendix, we present additional details regarding the performance evaluation of the proposed classifier, namely, the composition of the datasets and the metrics used for the evaluation. Additionally, we also include the summary of the analysis of the computational resources consumption.

The datasets used for the performance evaluation were collected in a testbed, which made it possible to save information about the application that generated each flow. The applications and services used during the capture of the datasets were running in *Microsoft Windows* or *Linux* operating systems and belong to different types of applications.

Table 2 describes the percentage of each type of traffic. The share of traffic from P2P applications in the three datasets spans from 55% to 65% in packets and from 40% to 50% in bytes. Since we addressed the classification

TABLE 2
Composition of the datasets used in the performance evaluation.

Traffic	Dataset 1						Dataset 2						Dataset 3					
	Packets (%)			Bytes (%)			Packets (%)			Bytes (%)			Packets (%)			Bytes (%)		
		TCP	UDP		TCP	UDP		TCP	UDP		TCP	UDP		TCP	UDP		TCP	UDP
P2P file-sharing	31.475	29.383	02.002	28.817	27.420	01.357	32.143	29.944	02.112	28.449	27.238	01.175	28.752	26.338	02.329	26.751	25.308	01.417
P2P streaming	30.994	00.070	30.819	20.118	00.027	20.076	24.885	00.081	24.729	15.634	00.047	15.577	30.130	00.053	30.010	20.979	00.020	20.950
P2P VoIP	00.684	00.025	00.659	00.117	00.007	00.110	02.710	00.033	02.676	00.440	00.009	00.431	02.953	00.028	02.925	00.474	00.008	00.466
Mail	06.186	06.186	00.000	08.182	08.182	00.000	01.078	01.078	00.000	01.394	01.394	00.000	02.795	02.795	00.000	03.610	03.610	00.000
HTTP download	03.901	03.901	00.000	06.273	06.273	00.000	04.438	04.438	00.000	06.999	06.999	00.000	04.204	04.204	00.000	06.612	06.612	00.000
Web browsing	00.431	00.431	00.000	00.503	00.503	00.000	01.444	01.444	00.000	01.740	01.740	00.000	00.502	00.502	00.000	00.562	00.562	00.000
Streaming	13.505	13.504	00.000	17.832	17.832	00.000	20.065	20.063	00.000	26.920	26.919	00.000	15.066	15.064	00.000	19.550	19.549	00.000
Telnet / SSH	00.013	00.013	00.000	00.003	00.003	00.000	00.010	00.010	00.000	00.003	00.003	00.000	00.003	00.003	00.000	00.001	00.001	00.000
FTP / SFTP	12.810	12.518	00.292	18.154	18.105	00.049	12.733	12.733	00.000	18.239	18.239	00.000	15.159	15.159	00.000	21.345	21.345	00.000
Online gaming	00.000	00.000	00.000	00.000	00.000	00.000	00.493	00.490	00.000	00.183	00.182	00.000	00.435	00.412	00.019	00.116	00.111	00.005

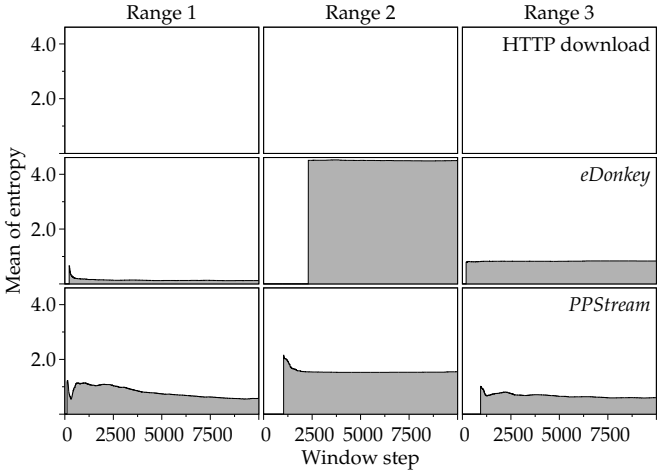

Fig. 4. Mean of the entropy, for all traffic, in three ranges of packet lengths for three additional examples, using a sliding window with size of 100 packets.

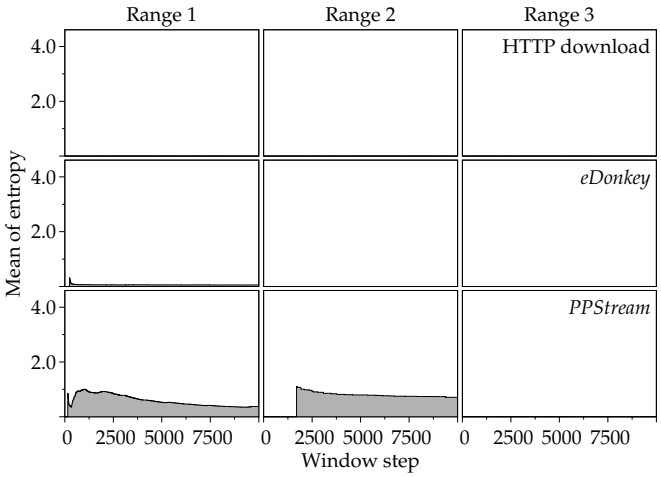

Fig. 5. Mean of the entropy, for outgoing traffic, in three ranges of packet lengths for three additional examples, using a sliding window with size of 100 packets.

of P2P Voice over Internet Protocol (VoIP) traffic in a previous work, herein we focus mainly on P2P streaming and P2P file-sharing. Therefore, the percentage of P2P

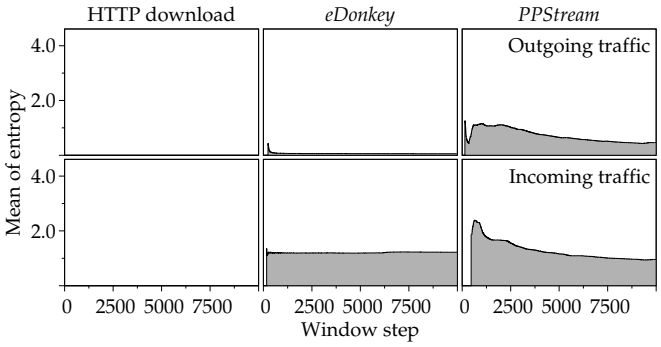

Fig. 6. Mean of the entropy for incoming and outgoing traffic for three additional examples, using a sliding window with size of 100 packets.

VoIP was much lower. The first dataset does not include traffic from on-line gaming. However, it contains more data from mail protocols than the other datasets.

The performance of the classifier was evaluated using three metrics: precision, recall, and accuracy. These metrics are defined in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) cases, as formalized by the following expressions [1], [2]:

$$Precision = \frac{TP}{TP + FP} \quad , \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \quad , \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad . \tag{3}$$

In the main article, we presented the performance results of four traffic classifiers: *Blinc*, *C4.5* decision tree, Support Vector Machines (SVMs), and Naïve Bayes. Table 3 contains the results for four additional classifiers integrated in the *NeTraMark* platform: port-based, payload-based, bayesian networks, and neural networks. The port-based classifier presents a very high precision, which means that all the traffic classified as P2P is, in fact, P2P traffic. However, the recall shows that only a small percentage of all P2P traffic was correctly classified.

TABLE 3
Results of the performance evaluation for four additional classifiers.

Datasets	Port-based			Payload-based			Bayesian Networks			Neural Networks		
	Accur.	Precision	Recall	Accur.	Precision	Recall	Accur.	Precision	Recall	Accur.	Precision	Recall
Dataset 1	48.94%	100.00%	03.93%	48.84%	100.00%	03.74%	44.67%	47.20%	34.56%	45.62%	44.44%	09.27%
Dataset 2	57.80%	100.00%	05.81%	59.68%	79.16%	13.58%	53.93%	48.35%	41.45%	53.98%	43.08%	08.43%
Dataset 3	55.93%	100.00%	11.39%	51.24%	56.26%	08.84%	48.99%	48.50%	41.41%	52.68%	75.00%	07.30%

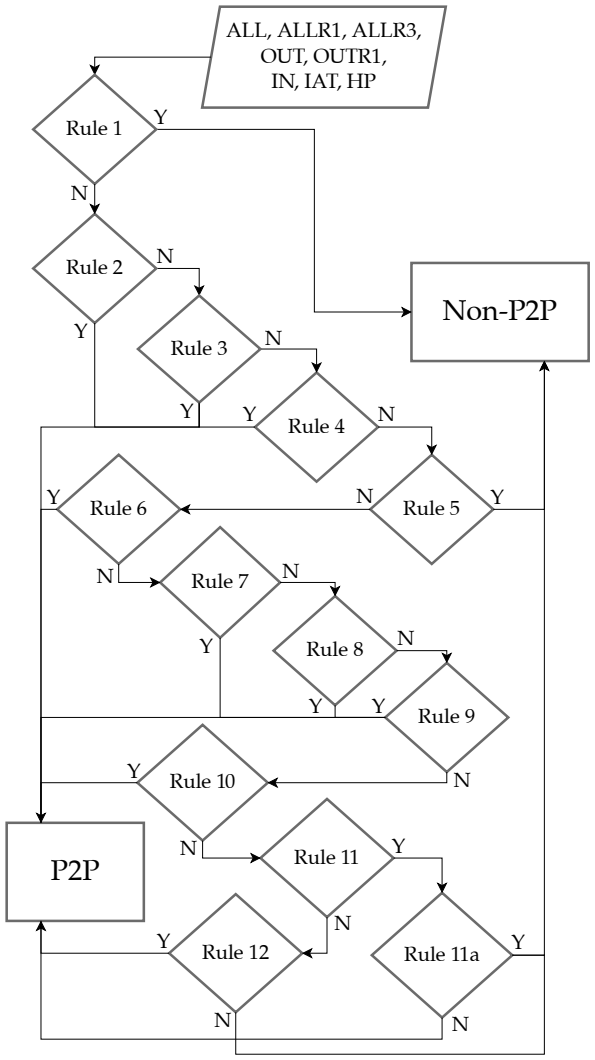

Fig. 7. Flowchart of the rules matching process of the proposed classifier.

TABLE 4
CPU time and maximum memory used by the classifier to process five distinct trace files.

Trace files	Packets	*host/port* pairs	CPU time (s)	Memory (KB)
File 1	500000	2324	41.60	38616
File 2	700000	3003	61.00	49152
File 3	1500000	5915	136.73	94340
File 4	1750000	6799	159.50	108348
File 5	2500000	9532	231.41	150688

In order to understand how the consumption of computational resources grows as the traffic increases, we made several measurements of the CPU and memory used by the classifier. We repeated the evaluation for trace files with a distinct number of packets and *host/port* pairs. As a complement of Fig. 12 in the main article, the results in Table 4 show that the CPU and memory consumption increases linearly with the volume of analyzed data.

References

[1] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proc. DARPA Broadcast News Workshop*, Feb. 1999, pp. 249–252.
[2] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*, 1st ed. Springer, Mar. 2008.

Chapter 7

Conclusions and Future Work

This chapter presents the main conclusions that result from the research work described in this thesis. Furthermore, it discusses a few research topics related with the work developed in the doctoral programme that may be addressed in the future.

1 Final Conclusions

This thesis is focused on the classification of traffic generated by peer-to-peer (P2P) applications and describes the research work developed with the purpose of presenting a new classification solution capable of identifying the traffic generated by P2P applications, in real-time, without using payload data. The research work was divided in four steps: the study of the problems raised by P2P traffic and the analysis of the existing approaches for traffic classification, the study of traffic immediately after its generation to analyze the traffic properties that result directly from P2P applications, the identification of unique characteristics of the traffic from hosts running P2P applications, and the classification of individual flows from P2P Voice over Internet Protocol (VoIP) sessions and from P2P media streaming and P2P file-sharing applications. Each of these four research steps resulted in contributions of this thesis, which ultimately contributes to the accomplishment of the main objective of developing a P2P traffic classifier that is able to operate in real-time and does not resort to payload data.

The emergence of the P2P paradigm increased the traffic load in the network edges and reduced the asymmetry between the incoming and outgoing traffic. Such fact raises concerns and challenges for Internet Service Providers (ISPs) and network administrators who required effective classification methods to manage the traffic load based on the nature of the traffic. However, classification approaches based on port numbers are now obsolete, while Deep Packet Inspection (DPI) generally requires great computational resources when used to process large amounts of traffic in real-time and is affected by the increasingly common payload encryption. Hence, several approaches based on traffic behavior have been proposed by researchers. Nevertheless, many of them are limited to offline use or to specific P2P protocols, or resort to complex algorithms, using several traffic features.

Therefore, the main goal of this thesis was to propose an alternative method that does not

suffer from the same limitations. The intermediate objectives were established so as to divide the research work needed to accomplish the main objective. The first part of the research work consisted in the study of the challenges raised by P2P applications and the analysis of the existing classification approaches. The adoption of evasive measures by P2P applications motivated the development of alternative classification methods. The literature review showed a clear trend towards the development of more effective behavioral methods. Nevertheless, since this kind of methods still cannot achieve the same accuracy of DPI-based classifiers, some authors are also proposing new and more efficient DPI methods that use less computational resources, so as to avoid the limitations of this approach. Moreover, some studies also used DPI in specific parts of the flows where the payload is not yet encrypted to identify traffic from applications that use encryption. Besides of this, it is not yet easy to evaluate which are the most promising approaches and the context where each of them are more effective as the studies in the literature lack consistency in the performance assessment of the methods they are proposing. The metrics used in the evaluations made by the authors are not always the same and in many cases are insufficient, but more important is the fact that each proposed method is evaluated using different traffic traces. The major reason for this problem is the scarce solutions to assess the ground-truth information and the privacy issues that make the datasets sharing troublesome.

The second part of the research work was described in chapter 3 and encompassed the study of traffic at its source. The study of the traffic generated by hosts running a single P2P or non-P2P application revealed characteristics that differentiate both types of traffic. The *chaotic* nature of P2P traffic, resulting from several parallel connections, influenced the analyzed traffic features. This influence was more easily observed in the packet lengths of P2P and non-P2P traffic. The packet lengths of the aggregated traffic of a host running a single non-P2P application, generally, presented a bimodal distribution with two more common lengths, whereas the packet lengths from P2P were very heterogeneous, with many distinct lengths being observed in the aggregated traffic from a host running a single P2P application. This observation strongly influenced the course of the research work, which explored mostly this property to classify P2P traffic.

The heterogeneity of the packet lengths was further studied and explored in the third part of the research work and more and larger traffic traces from an extended set of P2P and non-P2P applications were analyzed. The heterogeneity of the packet lengths in the traffic generated by a host running a P2P media streaming or P2P file-sharing application results from the aggregation of several parallel flows used to share contents with other peers, and by the search mechanisms used to find contents. In the case of VoIP applications, the packet lengths are more varied as a consequence of the Variable Bit Rate (VBR) speech codecs that are used, in most sessions, by applications like *Skype* or *Google Talk*. The heterogeneity of the packet lengths was measured

by resorting to entropy and to a sliding window with a constant size of *N* packets. The analysis was repeated for different values of *N*, showing that entropy is more stable when the size of the window is larger, though it takes more time to fill the window for the first time. As a compromise between stability and the time need to fill the window, a value between 100 and 500 packets was used, depending on the purpose of the method. The entropy results obtained for the traffic from different applications were compared, showing that the packet lengths from VoIP traffic generate a very high entropy, while the entropy for P2P media streaming and P2P file-sharing is lower but still higher than the one obtained for non-P2P applications, which is very low.

The behavior mentioned in the previous paragraph was observed in the traffic generated by a single application running in a host. Since most hosts run more than one application at the same time, in order to use the entropy of the packet lengths to identify hosts running P2P applications, it was necessary to analyze the effect in the entropy of running several applications in a host. The results showed that the presence of P2P traffic is still noticeable in the entropy when a P2P application is running together with other non-P2P applications. Moreover, it was also necessary to verify if the aggregated traffic from several non-P2P applications running in the same host does not raise the entropy to values that may be erroneously interpreted as an indicator of P2P traffic. In fact, in some extreme cases, the concurrent use of several non-P2P applications generates a high entropy of the packet lengths. To avoid this problem, the entropy was also computed separately for outgoing packets, and using intervals of 200 bytes so that all packet lengths belonging to the same interval are used in the entropy computation as being equal lengths. These two additional analyses enabled the differentiation of P2P and non-P2P traffic using only the packet lengths. A host-based classifier was proposed using only four rules that verify the entropy in the three distinct analyses. During the evaluation performance, the proposed classifier was able to identify the traffic from hosts running P2P applications with a false positive rate ranging from 0.00% to 10.42% and a false negative rate between 07.69% and 12.50%.

The fourth part of the research work, which was described in chapters 5 and 6, included the classification of individual flows from P2P VoIP, P2P media streaming, and P2P file-sharing applications. As observed previously in the research work, the three types of P2P applications generate packets whose lengths are very heterogeneous when compared to the ones from non-P2P applications. Nonetheless, the heterogeneity has different causes for P2P VoIP and for P2P media streaming and P2P file-sharing applications. In the case of the traffic from VoIP sessions, the heterogeneity is observed in each individual flow that is used in each session, which would make it easily identifiable by using the same entropy-based approach that was previously proposed. However, this happens because applications like *Skype* or *Google Talk* preferably use VBR codecs, even if they also support Constant Bit Rate (CBR) codecs. If a CBR speech codec is

used in a VoIP session, the effect in the heterogeneity and in the corresponding entropy will be the opposite, as the packet lengths will be extremely homogeneous.

Since the analysis of individual flows was focused initially in VoIP session flows, the traffic from many VoIP sessions using different VBR and CBR speech codecs and VoIP applications was analyzed to understand the properties of the packet lengths. The set of speech codecs considered in the study is supported by several available VoIP applications. Moreover, all versions of *Skype*, available at the time the research work was performed, were tested and used in experimental activities to understand which codecs were used during the different stages of development of the client application and which of their differences are reflected in the packet lengths. The analysis showed that the packet lengths in VoIP sessions depend mainly on the speech codec used by the application and that most speech codecs generate packets whose lengths are contained in different intervals and originate distinct levels of entropy.

The intervals of packet lengths and entropy observed for each speech codec enabled the definition of a set of behavioral signatures formed by both intervals and associated to a speech codec. The behavioral signatures were used in a classification method to classify VoIP traffic and identify the speech codec used in each session. The proposed method processes only one traffic feature, the packet length, and uses it together with the entropy to obtain the classification results. Additionally, the same approach based on a sliding window that was previously used in this thesis was also included in this method to make it suitable for real-time operation. In the performance evaluation, the proposed VoIP classifier was able to correctly classify the VoIP traffic with a sensitivity between 92.31% and 100.00% and a specificity between 99.51% and 99.99%. Moreover, the speech codec used in the VoIP sessions was identified with a sensitivity between 70.00% and 93.34% and a specificity between 99.99% and 100.00%.

Unlike what happens with the VoIP traffic, the heterogeneity of the packet lengths in the aggregated traffic of a host running P2P media streaming or P2P file-sharing applications results from the aggregation of several parallel flows used to share contents with multiple peers. In these cases, the heterogeneity of the packet lengths from individual flows is not easily distinguishable from the heterogeneity observed in non-P2P flows. Hence, the entropy was separately computed and analyzed for the packet lengths that fall into three distinct ranges of lengths: from 0 to 100 bytes, from 101 to 900 bytes, and from 901 to 1500 bytes. This separate analysis made it possible to explore the differences between the packet lengths from P2P and non-P2P traffic observed in these specific ranges. The packets in the first range are used by P2P applications to search for contents and answer to requests from other peers, whereas the packets from non-P2P applications that belong to the same range are mostly used to send acknowledge messages. Also in the third range, the packets from P2P applications have generally more heterogeneous lengths than the ones from non-P2P applications. Additionally, this analysis was also separately performed for incoming and outgoing traffic. The computation of the entropy of

the packet lengths for the different ranges was sufficient to correctly characterize the majority of the experimental traffic. Nevertheless, to improve the accuracy of the classification, the heterogeneity of two additional traffic features was also analyzed: the inter-arrival times with a precision of 0.1 seconds, and the remote *host*/*port* pairs.

Similarly to the method used for VoIP traffic classification, the entropy was computed by resorting to a sliding window for each flow processed by the classifier, so that it could be used in real-time operation. In each iteration of the sliding window, the mean of the entropy since the first iteration is computed and analyzed. The entropy mean results obtained in the analysis of the heterogeneity of the packet lengths, inter-arrival times, and remote *host*/*port* pairs by resorting to entropy were used to define a set of rules to classify P2P traffic. The majority of those rules uses only the entropy of the packet length feature, while all the other rules use the entropy of the packet lengths and inter-arrival times, with the exception of one rule that uses the entropy of the remote *host*/*port* pairs. In the performance evaluation, the proposed classification method was able to correctly identify between 92.90% and 94.50% of the total amount of P2P traffic, between 88.03% and 91.53% of the P2P file-sharing traffic, and between 99.81% and 99.90% of the P2P media streaming traffic. The precision of the method ranged from 97.52% to 98.94%.

The different analyses of the traffic generated by P2P applications performed during the research work described in this thesis showed that the distributed nature of the P2P paradigm influences the traffic, whose behavior is less predictable and more difficult to characterize. The multiple parallel connections to other peers, used to share several types of contents, following different physical links, possibly with distinct Maximum Transmission Units (MTUs), and passing through networks with different traffic management policies affect the properties of the traffic from P2P applications, which presents a *chaotic* behavior in terms of packet lengths. The contribution of this thesis for exploring this *chaotic* behavior by using the entropy to measure the heterogeneity of traffic features is the basis of the other contributions presented herein. The entropy results obtained in P2P and non-P2P traffic, especially for the packet lengths, but also for the inter-arrival times and remote *host*/*port* pairs, enable the discrimination between both types of traffic. Together with a sliding window containing a constant number of packets, the computation of the entropy can be made in real-time and, therefore, the results can be used in traffic classifiers that operate in real-time.

The main objective of this thesis was accomplished by the presentation of the three classifiers. Together, the proposed classifiers enable the classification of the traffic from hosts running P2P applications, the flows used by P2P VoIP sessions, and the flows used by P2P media streaming and P2P file-sharing applications. By using only the heterogeneity of traffic features from the packet headers, the classifiers can be used to classify traffic whose packet payloads are encrypted, while the computation of the entropy in an iterative manner allows the method

to be used in real-time. Furthermore, the classification of the traffic as generic P2P traffic, instead of focusing on specific P2P protocols or applications, makes the method capable of identifying the traffic from emergent or unknown protocols. In fact, during the research work, it was possible to classify as P2P the traffic generated by the video streaming in a web-site. Although this was initially considered a false positive case, after verification it was possible to confirm that the website was, indeed, using a plugin that allows it to use the P2P paradigm to reduce the costs of video distribution.

2 Future Work

The analysis of the heterogeneity of traffic features may be further developed and also used in cooperation with other approaches. Although the complexity of the classifiers is generally undesirable as it may increase the required computational resources, the integration of different classification approaches in the same classifier, working in cooperation towards the most accurate and useful results, may offer interesting possibilities by taking advantage of the best of each approach.

A cooperation scheme that would be easily implemented could include a first module performing the deep inspection of the packets, benefiting from the better accuracy of DPI methods. Using a second module performing the analysis of the entropy of traffic features would enable the classification of traffic containing encrypted payloads or generated by emerging or unknown protocols, for which the DPI module does not contain payload signatures. The entropy-based module could also be used to perform the analysis of the traffic from predefined port numbers that are sometimes used to disguise traffic from P2P applications or to verify some of the results obtained with the DPI module. For example, applications like *Gnutella* use the Hypertext Transfer Protocol (HTTP) to search for users and contents and, thus, the traffic they generate is sometimes simply classified as HTTP or even Web traffic. In such cases, the use of an entropy--based module would help to reveal the true nature of the traffic that was classified with the DPI module.

In order to implement an effective classifier based on this cooperation scheme, it would be necessary to perform a detailed study of the types of traffic that are more often erroneously classified and of what would be the gain, in terms of efficiency, of using an entropy-based module only for specific ranges of port numbers, so as to identify applications that are trying to hide their traffic using ports associated with other protocols, like HTTP or Secure Shell (SSH). Nevertheless, although this cooperation scheme may offer advantages for the classification accuracy, it would not solve the problem of the computational resources used by DPI methods.

Using the DPI and entropy-based modules in the opposite order, would allow the classifier to obtain gains in terms of resources efficiency. One of the reasons why DPI methods are generally computationally demanding is the fact that they have to check a large number of signatures and try to match them with each packet payload. Therefore, the list of signatures used by the DPI method could be divided in different groups, each of them associated with different entropy values for several traffic features. By doing so, it would be possible to first analyze the traffic with the entropy-based module to identify the group of payload signatures that would then be used by the DPI module. Such cooperation approach would, possibly, reduce the resources used by the DPI module, increasing the efficiency of the classifier.

Nonetheless, a cooperation scheme that uses the DPI module after the entropy-based module raises a few important challenges. The real-time entropy analysis, as proposed in this thesis, requires the use of a sliding window with a size of *N* packets. Only after the first *N* packets, it is possible to obtain the first entropy value. If the entropy analysis is performed for each flow, the classifier would have to keep the entire first *N* packets in the memory, so that, in case the DPI method is used, the payloads of the previous packets are still available.

The above problem can be minimized by limiting the packets kept in memory to a small number of the first packets or to specific packets in the flow. To do so, it would be necessary to study the flows from different types of applications and verify which are the most *valuable* packets for DPI. The conclusions obtained in such study could possibly show that the most *valuable* packets differ for each group of payload signatures, in which case the classifier would keep in memory different packets of the flow depending on the results previously obtained by the entropy-based module. The study of the different possibilities of cooperation between the different classification approaches would also result in the presentation of a classifier that could be made available for other researchers.

The entropy-based classification can also be used to perform a higher level classification of the traffic. Instead of being classified as generated by P2P or non-P2P applications (or even any finer granularity classification), the traffic can be classified simply based on the heterogeneity of specific traffic features, without associating it with any applications. Such method would be used for establishing different traffic management and routing policies. The management of the traffic in the network or even its routing would be based, or at least consider the behavioral properties of the traffic given by the analysis of the entropy of different traffic features. Although the benefits obtained with such approach are not clear, it would be interesting to study the information that can be extracted from the traffic by resorting to this analysis. This study would have to included the evaluation of the information required to defined traffic management policies in computer networks and also of which information revealed by the entropy analysis of the traffic features might be applied in the improvement of network and traffic management tasks.

In the same line of research work, it would also be possible to consider the analysis of the heterogeneity of several traffic features for the characterization of network traffic. This type of analysis could be used in cooperation with other methods and metrics typically used to characterize the traffic in computer networks. In this context, it may be interesting to perform a study on the feasibility of such analysis and of the information obtained using the heterogeneity analysis that could be valuable to characterize traffic in computer networks.