Issues and Future Directions in Traffic Classification

Article *in* IEEE Network · January 2012

DOI: 10.1109/MNET.2012.6135854 · Source: DBLP

3 authors, including:

Antonio Pescapè
University of Naples Federico II
194 PUBLICATIONS **3,063** CITATIONS

SEE PROFILE

Issues and Future Directions in Traffic Classification

Alberto Dainotti and Antonio Pescapé, University of Napoli Federico II
Kimberly C. Claffy, University of California San Diego

Abstract

Traffic classification technology has increased in relevance this decade, as it is now used in the definition and implementation of mechanisms for service differentiation, network design and engineering, security, accounting, advertising, and research. Over the past 10 years the research community and the networking industry have investigated, proposed and developed several classification approaches. While traffic classification techniques are improving in accuracy and efficiency, the continued proliferation of different Internet application behaviors, in addition to growing incentives to disguise some applications to avoid filtering or blocking, are among the reasons that traffic classification remains one of many open problems in Internet research. In this article we review recent achievements and discuss future directions in traffic classification, along with their trade-offs in applicability, reliability, and privacy. We outline the persistently unsolved challenges in the field over the last decade, and suggest several strategies for tackling these challenges to promote progress in the science of Internet traffic classification.

The variety and complexity of modern Internet traffic exceeds anything imagined by the original designers of the underlying Internet architecture. As the Internet becomes our most critical communications infrastructure, service providers attempt to retrofit functionality, including security, reliability, privacy, and multiple service qualities, into a "best effort" architecture originally intended to support a research environment. In order to prioritize, protect, or prevent certain traffic, providers need to implement technology for *traffic classification*: associating traffic flows with the applications — or application types — that generated them. When the focus is on detecting specific applications (e.g., Skype), the term *traffic identification* is sometimes used. Despite the increasing dependence on the Internet, there is essentially no scientifically reproducible body of research on global Internet traffic characteristics due to the sensitivity of and typical restrictions on sharing traffic data. Despite these constraints, security concerns and economic realities have motivated recent advances in traffic classification capabilities. Situational awareness of traffic is essential to prevention, mitigation, and response to new forms of malware, which can suddenly and rapidly threaten legitimate service on network links. Arguably as important, the high cost of deploying and operating Internet infrastructure compels providers to continually seek ways to optimize their network engineering or otherwise increase return on capital investments, including application-based service differentiation and content-sensitive pricing.

For these reasons, the state of the art in traffic classification has experienced a major boost in the past few years, measured in the number of publications and research groups focused on the topic. Diverse interests have led to a heterogeneous, fragmented, and somewhat inconsistent landscape. A recent survey of traffic classification literature reviewed advantages and problems with different approaches, but acknowledged their general lack of accuracy and applicability [1], whereas others took a narrower focus, taxonomizing and reviewing documented machine-learning approaches for IP traffic classification [2].

In this article we provide a critical but constructive analysis of the field of Internet traffic classification, focusing on major obstacles to progress and suggestions for overcoming them. We first give an overview of both the evolution of traffic classification techniques and constraints to their development. After briefly summarizing results of surveys in this field, we highlight key differences across existing approaches and techniques. We then discuss the main obstacles to progress in the current state of the art, including required trade-offs in applicability, reliability, performance, and respect for privacy. The persistently unsolved challenges in the field over the last decade suggest the need for different strategies and actions, which we recommend in the concluding section.

Traffic Classification: Evolution and State of the Art

At least three historical developments over the last two decades have rendered less accurate the traditional method of using transport-layer (TCP and UDP) ports to infer most Internet applications (*port-based approach*):

- The proliferation of new applications that have no IANA registered ports, but instead use ports already registered (to other applications), randomly selected, or user-defined
- The incentive for application designers and users to use well-known ports (assigned to other applications) to disguise their traffic and circumvent filtering or firewalls
- The inevitability of IPv4 address exhaustion, motivating pervasive deployment of network and port address translation,

 0890-8044/12/$25.00

where, for example, several physical servers may offer services through the same public IP address but on different ports

Despite its inaccuracy, associating transport layer ports with specific applications is still the fastest and simplest technique for continuous monitoring and reporting, often used operationally when accuracy is not critical.

As application design and user behavior rendered port-based flow classification unreliable, payload-based approaches emerged, which inspect packet content to identify byte strings associated with an application, or perform more complicated syntactical matching. The most common payload-based approaches compare packet content (payload) to a set of stored signatures (*pattern matching*), implemented in open source[1] as well as proprietary[2] tools. Payload examination is considered a reliable technique for Internet traffic classification, but poses formidable privacy challenges — privacy policies and laws may prevent access to or archiving of packet content. Payload inspection technology — sometimes called *deep packet inspection* (DPI) — also face technological and related economic challenges: it is easily circumvented by encryption, protocol obfuscation or encapsulation (e.g., tunneling traffic in HTTP), and prohibitively computationally expensive for general use on high-bandwidth links. These concerns with DPI techniques have motivated researchers to seek new discriminating properties of traffic classes and other classification techniques that do not require payload examination. Algorithms from the pattern recognition field using machine-learning techniques have proven promising, especially in the face of obfuscated and encrypted traffic which precludes payload analysis. These systems learn from empirical data to automatically associate objects with corresponding classes. In supervised algorithms, the classes are defined by the researcher, and the sample objects are given to the system already labeled with classes; whereas in unsupervised algorithms, the system identifies distinct classes and assigns objects to these classes (e.g., clustering). Many Internet applications generate traffic with specific characteristics amenable to classification using machine learning. In fact, supervised machine-learning approaches have achieved results comparable to DPI [2]. Unsupervised machine-learning techniques are a promising way to cope with the constant changes in network traffic, as new applications emerge faster than it may be possible to identify new signatures and train machine-learning classifiers. The performance of such classifiers depends not only on the differences among machine-learning algorithms (neural networks, decision trees, Bayesian techniques, etc.) and their specific configuration, but also on the selection of classification features, which are the types of data used to "describe" each object to the machine-learning system. Features include common flow properties (e.g., per-flow duration and volume, mean packet size) as well as more detailed properties, such as sizes and interpacket times of the first *n* packets of a flow, or entropy of byte distribution in packet headers or payload. Identifying particular traffic classes or applications (e.g., VoIP or Skype) requires discerning even more specific features, and must contend with application software changes, including those designed to preclude classification.

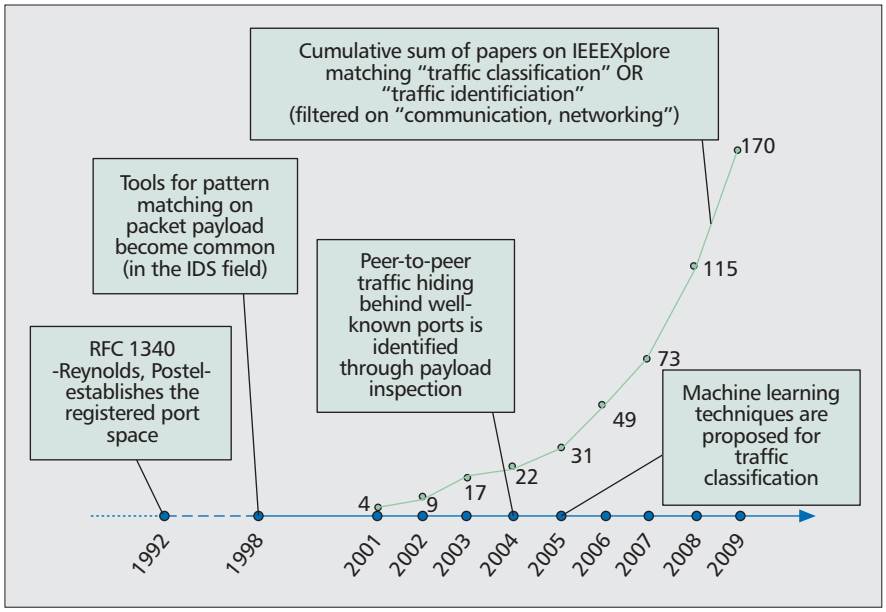

Figure 1. *Evolution of approaches and literature in traffic classification.*

Finally, approaches based on host communication patterns use heuristics that can effectively complement payload inspection techniques, especially for obfuscated traffic. For example, keeping a table of (*IP*, *Port*) pairs for each flow classified by payload inspection allows identification of unclassified flows that have a source or destination IP stored in the table [3, 4]. Another approach tries to identify peer-to-peer applications by correlating the social network of a given host with its transport-level interactions [5]. Unfortunately, this approach requires seeing both directions of each traffic flow, so it can only be used on single-homed edge or near-edge links.

The evolution of traffic classification technology (Fig. 1) has created a heterogeneous landscape, recently summarized in survey papers [1, 2]. These surveys taxonomize the available techniques by their classification algorithm, (e.g., port-based, DPI, machine-learning), and document the decreasing reliability of the port-based approach [6] and the ability of machine-learning approaches to achieve results comparable to far more privacy-invasive DPI techniques. However, surveys also show that a large portion of network traffic is still left unclassified by all techniques [1, 6]. Moreover, the literature exhibits a wide range of inconsistent terminology to describe approaches and metrics, making it difficult or impossible to compare studies or safely infer conclusions. While the existing surveys highlight inconsistencies in terminology and *evaluation metrics* [1, 7], here we draw attention to even more substantial differences: there is a wide methodological range of granularity in definitions of flows and traffic classes across approaches that makes different approaches difficult to systematically compare, even when using the same reference data and tools as well as the same evaluation metrics. The granularity of traffic flows reflects the portion of the packet headers analyzed to construct what we call *flow objects*, which can vary from one direction of an individual application session to bidirectional flows between hosts. Common flow objects, with different granularities, include:

- *TCP connections*: Heuristics based on the observation of some TCP flags (i.e., SYN, FIN, RST), or TCP state machines, are used to identify the start and the end of each connection.

[1] *http://l7-filter.sourceforge.net, http://www.bro-ids.org*

[2] *Cisco's NBAR, Juniper's Application Identification, QOSMOS' Deep Packet Inspection*

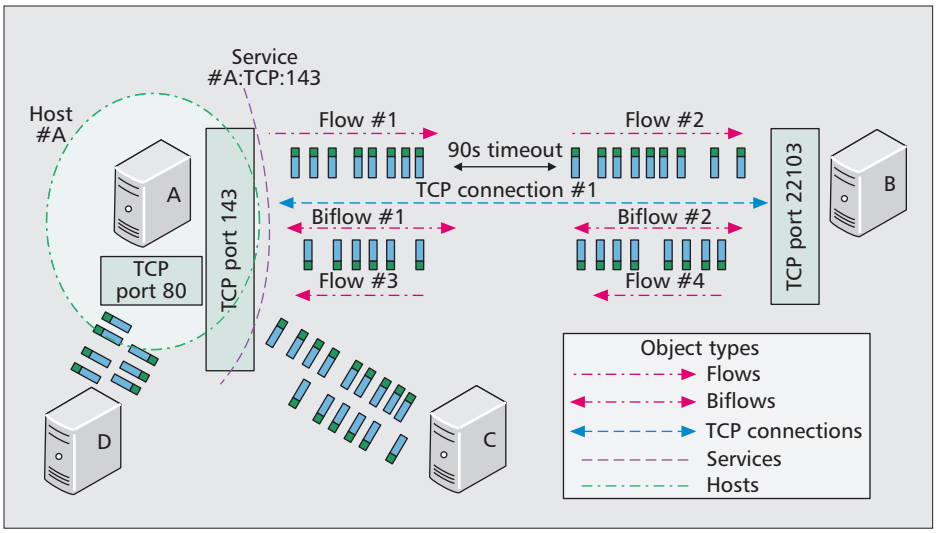

Figure 2. *Types of Flow Objects. In this example packets between hosts A and B can be grouped either into a single TCP connection, or two biflows, or four flows. The same packets can instead (or also) be part of a larger object that groups all packets to/from port TCP 143 of host A into a single service, i.e., including packets to/from host C in the figure, or part of a host, which groups all packets to/from host A, i.e., all packets in the figure.*

- *Flows*: A typical flow definition uses the 5-tuple {$source_{IP}$, $source_{port}$, $destination_{IP}$, $destination_{port}$, *transport-level* protocol}; some tools also use a flow timeout (60 s or 90 s of idle time to delineate the end of a flow) or periodic reset (e.g., timeout all flows on a 5-min boundary).
- *Bidirectional flows (biflows)*: Same as above, but includes both directions of traffic, assuming both directions of flows can be observed (especially challenging on backbones where Internet routing is often asymmetric). Classification approaches using bidirectional flows cannot be applied "as is" to flows or TCP connections because the classification features can change.
- *Services*: Typically defined as all traffic generated by an IP-port pair.
- *Hosts*: Some approaches classify a host by the predominant traffic it generates, assuming both directions of traffic (to and from the host) can be observed.

Furthermore, different approaches may ascribe flow objects to traffic classes of different size or granularity, such as:

- *Traffic profiles* (bulk, interactive, etc.)
- *Application categories* (e.g., chat, streaming, web, mail, file sharing)
- *Applications* (e.g., KaZaa, Edonkey, IMAP, POP, SMTP)
- *A single application vs. the rest* (i.e., identification)
- *Content type*, either coarse-grained (e.g., text, binary, or encrypted content) or fine-grained (e.g., text, picture, audio, video, compressed, base64-encoded image, base64-encoded text)

Figure 2 illustrates several different types of flow objects, the proper selection of which often depends on the purpose of classification (e.g., traffic management, security).

Obstacles and Future Directions in Internet Traffic Classification

Using the terminology and context provided in the previous section, we outline the persistently unsolved challenges in the field over the last decade, and suggest several strategies for tackling these challenges to promote progress in the science of Internet traffic classification.

Available Data and Ground Truth

The most obvious obstacle to progress on traffic classification is a persistent problem of Internet research generally: lack of a variety of *sharable traces* to serve as test data as well as *ground truth* (i.e., annotated flow objects used as reference) for validation. Balancing individual privacy against other needs, such as security, critical infrastructure protection, or even science, has long been a challenge for law enforcement, policymakers, and scientists. It is good news when regulations prevent unauthorized people from examining the contents of your communications, but current privacy laws often make it hard — sometimes impossible — to provide researchers with data needed to scientifically study the Internet. Our critical dependence on the Internet has rapidly grown much stronger than our comprehension of its underlying structure, performance limits, dynamics, and evolution, and unfortunately, current privacy law is part of the problem: legal constraints intended to protect individual communications privacy also leave researchers and policymakers trying to analyze the global Internet ecosystem essentially in the dark. Traffic classification is but one casualty.

One potential solution would be to share traces that are sufficiently aged as to have minimal privacy sensitivities, but since all classification tools must also contend with the application obfuscation arms race, the most relevant and formidable challenge is accurately classifying a substantial fraction of traffic on recent traces [6].

To address the difficulty in sharing even anonymized data, one proposed but untested and scientifically problematic alternative is to "*move the code to the data*," where researchers send their analysis tool (generally software) to a data provider who runs the tool against private data and sends the results back to the researchers. Several researchers independently proposed this model years ago, but there has been no measurable traction in this direction, partly because few data providers have the resources and incentive to review researcher software to ensure it will not leak unexpected information from the data.

Researchers have also explored the possibility of sharing anonymized traffic traces annotated with ground truth obtained via payload examination before anonymization [7]. Unfortunately, tools for labeling traces with ground truth are still in early development, do not consistently assign the same flow object to the same class [8], and most of them are not publicly available, so they cannot be scientifically evaluated or improved by researchers. Primarily based on matching the presence of known strings ("signatures") in the packet payload, these tools differ not only in the set of signatures, but also in the matching techniques and algorithms. For example, the *L7-Filter* tool, used in several studies (e.g., [9]), is strictly based on regular expressions applied to a portion of the payload stream (e.g., first 4096 bytes), while the *crl_pay* tool by Karagiannis *et al.* [4–6] limits the payload analysis to the first 16 bytes, but also uses port numbers and packet sizes to infer the generating application. Another problem is that some sig-

natures are too general (e.g., based on too few bytes) and can generate incorrect annotations. Most recently (2010), researchers have experimented with gathering ground truth directly from hosts of users volunteering to self-annotate their traffic, using an admittedly small population of (about 20) users.[3] Although still meager in scope, such technical developments reflect growing awareness by researchers of the need for accurate publicly available tools for ground truth annotation, as well as standard techniques, procedures, and annotated data sets to use as ground truth reference resources.

Traffic Evolution

Both research and marketing literature in traffic classification suggest there is no perfect classification technique (i.e., with 100 percent accuracy over all traffic. In addition to the three historical developments reviewed earlier (non-standard ports, disguised ports, and NATs) that have increased the difficulty of classifying traffic by port identifiers over the last two decades, three more recent trends this decade have further hindered the ability to classify Internet traffic:

- *Protocol encapsulation*, such as traffic tunneled inside HTTP, accurate identification of which requires more invasive payload inspection and/or complex protocol analysis in the classifier.
- Some traffic is encrypted or encoded, limiting the ability to extract features to those that remain after encryption.
- *Some applications support multiple service channels*: multichannel applications that merit different service qualities or security policies within the application require identifying not only the network application associated with a traffic flow, but also the specific task within the application (e.g., signaling, video streaming, chat, data transfer, voice call).

Traffic classification techniques in the literature have not kept pace with these three challenging trends.

Scalability

Another challenging trend in Internet evolution is the tremendous growth of the infrastructure in every dimension, including bandwidth capacity of links. Most real-world applications of traffic classification require tools to work online, reporting live information or triggering action according to classification results. But online traffic classification on modern links requires *trade-offs among accuracy, performance, and cost*. The practical challenges have led to many published studies with limited evaluation in a simplified environment rather than a systematic rigorous analysis of these trade-offs. For example, in order to work online without custom (often prohibitively expensive) hardware, complex DPI classifiers must sacrifice functionality — either analyzing a shorter portion of the payload stream of each traffic flow, or simplifying their pattern matching approaches.

Machine learning techniques require similar compromises to lower or bound the latency of classification during online execution. Data reduction is generally implemented by limiting the number of packets of a flow [9, 10] used for extracting classification features. Computational overhead is limited by reducing the set of features [11] used to classify traffic, ideally using features that can be extracted with low computational complexity. Some features are not suitable for online classification because they are available only at the end of a flow, such as total transferred bytes.

In [11] the authors analyzed the computational complexity and memory requirements associated with typical traffic features in an online classification context. Selecting 12 features, where the maximum complexity is $O(n \times \log2(n))$ (for median packet size), they show that while features like *source and destination ports* or *number of bytes sent in initial window* have complexity $O(1)$, most features used for online classification (e.g., *variance of packet size*) have complexity $O(n)$, with n being the number of packets used to extract features. Limiting the number of packets used to extract features offers several benefits: lower feature extraction complexity; lower latency since classification can occur early in each traffic flow; and lower memory cost to maintain flow state during classification. Using a limited set of packet traces, some researchers have shown that four to five packets were enough to approach the maximum classification accuracy obtainable with a restricted set of features suitable for online classification [10, 11]. However, these compromises also make the techniques easier to evade, so designers must consider the specific objective of the online classifer when optimizing performance, e.g., the evasion likelihood matters more for security-related contexts than for quality of service.

Latency is also affected by the speed of the specific machine-learning algorithm. Studies of features for online traffic classification and real traffic traces have shown that the fastest techniques among those most commonly used are based on decision trees, specifically the C4.5 algorithm [11, 12].

Architectural design choices also influence these scalability trade-offs. In the next decade, traffic classification systems will have to be *redesigned to run on multicore hardware, targeting low-cost but highly parallel architectures*. General-purpose graphical processing units (GPGPUs) have introduced a new computing paradigm, allowing scientific and computationally intensive applications to achieve orders of magnitude in performance improvements with minimal hardware costs. Recent works have successfully applied GPGPUs to DPI for intrusion detection and traffic classification [13], using multiple cores to speed up regular expression matching. Although not yet applied specifically to traffic classification, redesigning generic machine learning algorithms as support vector machines in order to exploit multicore systems has yielded large scalability improvements. Parallelism can also be pursued at a higher layer of a traffic classification architecture, using multithreading to *pipeline* the typical execution sequence: packet filtering, packet classification (aggregation of packets into flow objects), feature extraction, activation of different traffic classifiers, flow object classification by each classifier, combination (see the "Combining Techniques" section), and output. Alternatively, replicating classification modules on different cores may enable per-flow load balancing to achieve even higher scalability.

Consistent Evaluation and Comparison Methods

Rigorous evaluation and comparison of techniques requires *standard testing and validation procedures and benchmarking metrics*. We described earlier the lack of convergence in terminology in the literature, which also extends to benchmarking metrics used to evaluate methods [7]. The generally accepted metric for evaluation is *overall accuracy*, the fraction of all flow objects correctly classified. We have previously recommended borrowing metrics from more mature fields, such as those used in other machine-learning classification problems [6] — *precision*, *recall*, and *F-measure*, calculated for each class separately to yield a deeper understanding of the classifier's performance than a simple overall accuracy metric provides. Precision is the ratio of objects properly attributed to a class over the total number of objects attributed to that class. Recall is the percentage of objects from a given class that are properly attributed to that class. F-measure is calculated as 2

[3] *http://www.ing.unibs.it/ntw/tools/traces*

× *precision* × *recall*/(*precision* + *recall*); this last metric is useful to rank and compare the per-class performance of different classification algorithms. *Byte accuracy*, which is the ratio of the sum of all bytes carried by the correctly classified flow objects to the sum of all bytes in the traffic considered, is rarely analyzed in Internet traffic classification literature, although it is arguably more relevant operationally, and mitigates the *class imbalance problem* (a machine-learning term) induced by high population variance, in this case across Internet flow sizes [14].

Benchmarking metrics are more effective if they *take into account the target application* of the classification approach and distinguish among granularities under evaluation. Different *cost functions* of basic metrics, including error handling, depend on the specific application context (e.g., traffic management, differential pricing, security), which renders it difficult to standardize evaluation metrics. Using the terminology from earlier, some techniques classify into only four broad profiles (interactive, bulk data, streaming, transactional); other tools group applications into categories (mail, web, peer-to-peer [P2P], etc.) [5]; others consider individual applications [10]. Automated and rigorous comparison of techniques would require both *standard flow object definitions* and *standard classes* at each layer, as well as standard mapping between layers (e.g., IMAP, POP, and SMTP are all in the mail category).

Finally, and related to the first challenge described above (available data), the *traffic* used for test and validation is typically limited to what is easy to collect or share. Traffic traces for validation are often necessarily extracted from a single or a few links, or from links too similar in nature (e.g., university access networks and home access networks (digital subscriber line [xDSL]), which inhibits the evaluation of the robustness of tools in the face of more realistically varying traffic. Data sets may only include a subset of traffic on a given link. UDP traffic is often ignored, despite its growth on the global Internet. Many traffic traces do not include both directions of traffic flow, preventing their use for techniques based on overall host behavior [5]. Data used for evaluation is often years old, while identifying current traffic types, especially malicious, is a more common goal for those deploying traffic classification technology. The effect of sampled traffic — often the only type of data available due to measurement performance constraints — on feature extraction and classification accuracy has not been systematically explored, while complete traffic traces of sufficient length for evaluation are unwieldy and costly to store, curate, and use.

Combining Techniques

Since different techniques perform better on some traffic classes, a system combining them — called a *multiclassifier system* — potentially achieves better accuracy than any single classifier. The machine learning community has recently developed multiclassifier systems based on intelligent *combination algorithms* that learn from historical behaviors of individual classifiers on the studied flow objects. Such systems can achieve higher accuracy than any single classifier, and are more robust to changes in the sample population, including the nature and mix of applications (*concept drift*). Network anomaly and intrusion detection applications have successfully used such multiclassification approaches, but traffic classification tools have only attempted simplified approaches, such as resorting to host-based heuristics or machine learning techniques only after payload inspection has failed. Researchers are only recently beginning to investigate more general and effective techniques [15] that use different classifiers on the same flow object and combine the results through algorithms based on either voting, Bayesian probability, Dempster-Shafer theory, or the behavior knowledge space (BKS). Although combining classifiers can *increase the computational complexity* of the process, it can also potentially reduce the amount of traffic information required for accurate classification (e.g., using two packets per flow rather than five), which can reduce the average classification time (latency). However, such algorithms also typically require additional information in the training phase, such as confusion matrices or BKS tables. A per-classifier confusion matrix lists in each cell (i, j) the percentage of objects of class i recognized by the classifier as belonging to class j. A BKS table similarly lists the probability of an object belonging to each class, for each possible combination of outputs from different classifiers. Obtaining the data to populate the confusion matrix or BKS table requires individually training and testing each classifier before training the combination algorithm. Nonetheless, assuming that the different classifiers in the combination can execute in parallel, the flexibility offered by combination classifiers facilitates the scalability trade-offs essential for online techniques.

Finally, adding confidence values to the output of individual classification algorithms may further improve the accuracy of multiclassifier systems. Many machine learning algorithms can associate a confidence value with the inferred class, while, as regards payload-based approaches, one can also derive confidence values for a given output class by analyzing pattern matching signatures [9]. Using confidence values in conjunction with multiclassification enables the implementation of classifiers that may improve precision by refusing to attempt classification (a *rejection option*) under specific circumstances.

Available Implementations

A few available tools are worth noting. The NetAI[4] tool does not directly perform traffic classification, but can extract a set of features from both live and stored traffic for use by a general-purpose machine learning classifier. The *Fullstats* utility developed at the University of Cambridge, United Kingdom, is also able to extract classification features from a traffic trace. The same research group released *GTVS*, a DPI-based tool able to assist researchers in manually inspecting and semi-automatically labeling traffic traces.[5] To our knowledge the only two traffic classifiers implementing machine learning techniques presented in the literature are Tstat 2.0[6] and TIE.[7] Tstat uses a customized machine learning technique based on a Bayesian framework with *packet size* and *interpacket time* as classification features to identify applications such as Skype and obfuscated P2P file sharing. Although Tstat's machine-learning-based classification is limited to a few applications, the tool allows the extraction of a large number of classification features. It also performs both payload inspection and machine learning classification online on live traffic, and can generate web reports with graphs of aggregated data.

TIE is a software platform for supporting the implementation of traffic classification techniques inside a unified framework made available to the research community. TIE exposes a simple application programming interface (API; in the C language) for the development of traffic classification plugins adopting either DPI, machine learning, or port-based tech-

[4] *http://caia.swin.edu.au/urp/dstc/netai*

[5] *http://www.cl.cam.ac.uk/research/srg/netos/brasil*

[6] *http://tstat.tlc.polito.it*

[7] *http://tie.comics.unina.it*

niques. The modular architecture supports traffic capture and filtering, packet aggregation at several granularities, feature extraction, as well as combination of classifiers. We developed and released TIE to support advanced features such as multi-classification and online classification as well as to facilitate consistent comparison of different techniques through a framework of well defined classes, flow objects, and metrics, addressing some of the recommendations made in the next section.

Summary and Recommendations

Research on Internet traffic classification has produced creative and novel approaches, but the landscape is foggy, fragmented, and inconsistent. In this article we provide a critical but constructive analysis of the field of traffic classification, including its historical context, which illuminates its achievements and obstacles to progress. A recurrent emergent theme of our investigation is the need for cooperative approaches to the science of Internet classification, and a recognition of the incentives as well as counter-incentives of industry stakeholders to contribute to the transition of Internet traffic classification from art to science. We outline both research and policy directions that could improve the capabilities and effectiveness of traffic classification systems, summarized in the following recommendations:

- Rigorous evaluation and comparison requires testing and validation of tools against recent and complete traffic traces, which will require navigating the persistent challenges (mostly policy, some technical) of sharing traffic data with researchers.
- The ever increasing speed of network links requires rigorous investigation of scalability trade-offs in traffic classification. Appropriate and novel designs for highly parallel low-cost architectures promise significant scalability improvements.
- Improving tools to annotate data with the actual traffic class (i.e., ground truth tools) can be done through sharing of algorithms and signatures in order to allow community contributions, comparisons, and validation, for example, by comparing the output of the annotating tools against 100 percent safe reference data.
- Traffic classification techniques and algorithms should be presented with rigorous empirically grounded analysis of efficiency and performance, using standard metrics comparing implementations running on diverse Internet traffic, including encapsulated, encrypted, and multichannel application flows of varying length.
- Research on multiclassifier systems is warranted, since they combine the benefits of different approaches to improve accuracy, flexibility, and speed, at some cost in computational complexity and possibly additional training data and time.
- Publications of open source implementations of real traffic classification systems for use in experiments would foster collaboration and promote convergence on standard definitions, procedures, and reliable evaluation of techniques.

Many of these problems are complex policy rather than purely technical problems, and advancing the field will require that the Internet research community learn how to navigate the conflicting incentive structure of the phenomena they are trying to study. In the short term, we can imagine several concrete actions that would promote progress: community standardization (e.g., through Internet Engineering Task Force [IETF] Requests for Comments [RFCs]) of definitions, data formats, and metrics for traffic classification and identification; holding traffic classification competitions in conjunction with networking conferences and workshops; creating public repositories of traces of recent traffic from real network links annotated with ground truth; and establishing a coordinated network of entities offering the execution of classification code on their traces (i.e., *send-code-to-the-data*) and documenting experiences in formats that allow comparison with alternatives.

References

[1] A. Callado *et al.*, "A Survey on Internet Traffic Identification," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 3, July 2009.
[2] T. T. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification Using Machine Learning," *IEEE Commun. Surveys & Tutorials*, vol. 10, no. 4, 2008, pp. 56–76.
[3] A. W. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," *Proc. PAM '05*, 2005, pp. 41–54.
[4] T. Karagiannis *et al.*, "Transport Layer Identification of P2P Traffic," *Proc. 4th ACM SIGCOMM Conf. Internet Measurement*, 2004, pp. 121–34.
[5] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," *Proc. 2005 Conf. Apps., Technologies, Architectures, and Protocols for Comp. Commun., ACM SIGCOMM '05*, 2005, pp. 229–40.
[6] H. Kim *et al.*, "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices," *Proc. 2008 ACM CoNEXT Conf.*, 2008, pp. 1–12.
[7] L. Salgarelli, F. Gringoli, and T. Karagiannis, "Comparing Traffic Classifiers," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 37, July 2007, pp. 65–68.
[8] M. Pietrzyk, G. Urvoy-Keller, and J.-L. Costeux, "Revealing the Unknown ADSL Traffic Using Statistical Methods," *Proc. 1st Int'l. Wksp. Traffic Monitoring and Analysis*, 2009, pp. 75–83.
[9] G. Aceto *et al.*, "PortLoad: Taking the Best of Two Worlds in Traffic Classification," *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.
[10] L. Bernaille, R. Teixeira, and K. Salamatian, "Early Application Identification," *Proc. 2006 ACM CoNEXT Conf.*, 2006, pp. 6:1–6:12.
[11] W. Li *et al.*, "Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema," *Computer Networks*, vol. 53, Apr. 2009, pp. 790–809.
[12] N. Williams, S. Zander, and G. Armitage, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 36, no. 5, Oct. 2006, pp. 7–15.
[13] G. Szabó *et al.*, "Traffic Classification over Gbit Speed with Commodity Hardware," *IEEE J. Communications Software and Systems*, vol. 5, 2010.
[14] J. Erman, A. Mahanti, and M. Arlitt, "Byte Me: A Case for Byte Accuracy in Traffic Classification," *Proc. ACM SIGMETRICS MineNet Wksp.*, June 2007.
[15] A. Callado *et al.*, "Better Network Traffic Identification Through the Independent Combination of Techniques," *J. Network and Comp. Apps.*, vol. 33, no. 4, 2010, pp. 433–46.

Biographies

ALBERTO DAINOTTI (alberto@unina.it) received his Ph.D. in computer egineering and systems from the Department of Computer Engineering and Systems of the University of Napoli Federico II, Italy, where he currently works as a post-doctoral researcher. His research interests fall in the areas of network measurements, traffic analysis, and network security.

ANTONIO PESCAPÉ [SM] (pescape@unina.it) is an assistant professor in the Department of Computer Engineering and Systems of the University of Napoli Federico II. He received his M.S. Laurea degree in computer engineering and Ph.D. in computer engineering and systems, both from the same university. His research interests are in the networking field with focus on Internet monitoring, measurements, and management, and network security.

KIMBERLY CLAFFY (kc@caida.org) leads the Cooperative Association for Internet Data Analysis (CAIDA) at the University of California, San Diego (UCSD), and is adjunct professor of computer science and engineering at UCSD. Her research interests include Internet (workload, performance, topology, routing, and economics) data collection, analysis, and visualization. She has a Ph.D. in computer