On the Characterization of Multi-Channel Applications

Walter de Donato
Dipartimento di Informatica e Sistemistica
University of Napoli, Italy
walter.dedonato@unina.it

Antonio Pescapé
Dipartimento di Informatica e Sistemistica
University of Napoli, Italy
pescape@unina.it

ABSTRACT

We are assisting to the evolution of new generation applications and services, progressively providing - through a single interface - more interactions among the users and between the users and the network. This is promoting the development of *multi-channel applications* (e.g. Skype, Cloud Computing Platforms, Facebook, ...) that are specifically designed to easily manage different services delivered on different channels, providing a single access point for the users. This work proposes an integrated multi-layer methodology for the analysis, characterization, and identification of multi-channel applications. A proof of its applicability is shown considering Skype as a case study.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Miscellaneous

General Terms

Measurement

Keywords

Multi-Channel Applications, Traffic Identification.

1. INTRODUCTION AND MOTIVATION

Nowadays a change of paradigm is happening in the world of telecommunications: in a highly heterogeneous and dynamic context as the Internet, the user is becoming the real fulcrum. We are assisting to a radical change from the *Network-Centric* view to the *User-Centric* view. The user increasingly takes an active role in the network, promoting *peer-to-peer* (P2P) and *many-to-many* interactions. The variety of devices, together with his mobility, makes today the user a real network "*micro-operator*", sharing his wide-band connection and providing both contents and network functionalities. We are therefore assisting to a shift toward the so-called *User-Centric Internet (UCI)*.

The transition to the UCI view is fostering the development of *multi-channel applications*. Such applications provide a single interface to perform heterogeneous activities, usually exploiting many communication channels. Since traditional approaches independently look at channels, the study, monitoring, and control of network traffic is becoming less and less effective [1]. These are the main causes: (i) working with multi-channel applications we have also to cope with the problems of recognizing traffic flows associated to the same application and associating them with specific activities (e.g. signaling, video streaming, voice, file transfer, ...); (ii) transport layer port numbers are often randomly chosen or reused for non standard protocols; (iii) there is a trend toward an extensive use of encryption, obfuscation and encapsulation in communication channels. Therefore, it is necessary to find new techniques and analysis methodologies purposely designed for the properties of emerging applications. For example, considering the relations between channels belonging to the same application can reveal behavioral patterns otherwise not visible: our approach starts from this assumption.

Characterizing multi-channel applications has implications in many networking fields: (i) network planning and dimensioning; (ii) service differentiation; (iii) content delivery; (iv) intrusion and anomaly detection. The identification of network traffic could be the main application of such result. It should help providing a better accuracy and rising the percentage of identified traffic. Moreover, it should allow to improve the granularity of the traditional approaches. For instance, an ISP providing both Internet access and telephony services could be interested in blocking or shaping only VoIP (Voice over IP) traffic pertaining to a specific competitor. With respect to a multi-channel application providing also voice calls (e.g. Skype), the ISP may be forced to block/shape all its traffic. Such decision could force many users to change provider, thus resulting in a monetary loss. Whereas, being able to discriminate application activities, allows to selectively apply rules to them.

2. THE PROPOSED METHODOLOGY

We propose the definition of a novel methodology for the characterization of multi-channel applications working at different abstraction layers. The methodology is based on a multi-layer traffic inspection and a decomposition approach, as depicted in Fig. 1, counting four layers: (i) ***host***, (ii) ***service***, (iii) ***biflow*** (bidirectional flow) and (iv) ***packet***. The *host* layer aggregates the whole traffic pertaining to a single host. The *service* layer groups together packets having the same transport protocol and IP address-port pair. The *biflow*[1] layer aggregates packets belonging to the same channel (i.e. having the same 5-tuple, where source and destination can be swapped). Finally, the *packet* layer looks at the properties of each packet (e.g. size, inter-packet time, payload, ...). According to this decomposition, a biflow cor-

[1]Source and destination roles are related to the first packet.

CoNEXT Student Workshop'09, December 1, 2009, Rome, Italy.
ACM 978-1-60558-751-6/09/12.

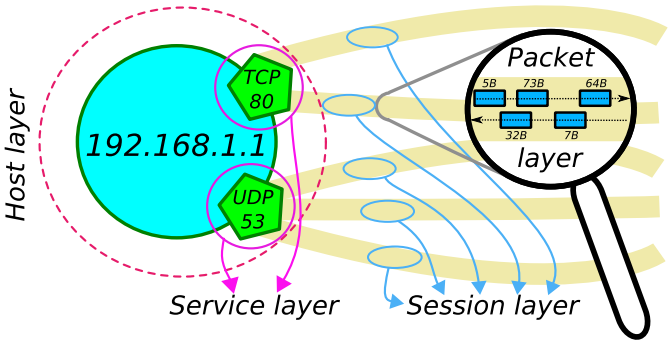

Figure 1: Analyzing traffic at different layers.

responds to a channel and, aggregating traffic at each layer, data is tipically inspected at lower layers (e.g. packet-sizes distribution at host layer).

Combining information collected at these layers can reveal useful patterns in host interactions, traffic flows statistics, congestion prevention/reaction mechanisms, overlay communications topologies, geolocalization aspects, etc. For instance, if the host in Fig. 1 is running eMule on TCP port 80 and UDP port 53 with obfuscation enabled, it would be difficult to identify it by independently looking at biflows exploiting port numbers, payload content or flow statistics. Whereas, characterizing the correlation between host and biflow layers could reveal patterns peculiar to the application (e.g. TCP/UDP biflows ratio, connections temporal sequences, ...). Therefore, correlating multiple channels has two main benefits: (i) by looking at many biflows belonging to the same application it is possible to detect the application itself; (ii) being aware of an application running on a particular host/service can help in associating a new flow to it, and to identify the related activity.

3. EXPERIMENTAL ANALYSIS: A PROOF OF CONCEPT

To prove the feasibility and the benefits of the proposed methodology, we applied it to Skype. Skype represents an interesting case study since it works on a super-peer based P2P overlay architecture, its communications are mostly encrypted and the adopted protocols are secret.

We used TIE[2] to gain knowledge of the traffic associated to each channel (see Tab. 1), and we discovered several patterns[2] at different layers. We found that, differently from traditional applications, Skype listens for both TCP and UDP connections on the same fixed port number[3], randomly chosen at installation time[4]. Moreover, when connected to the P2P network, it always has at least one persistent TCP connection with a super-peer.

[2]Since Skype exposes different patterns depending on network configuration, we present a preliminary analysis of the generic super-peer case: public IP address and no firewall restrictions.

[3]33837 represents such fixed port number

[4]It also listens on ports 443 and 80 to provide connectivity in presence of firewalls.

Table 1: Skype traffic at biflow layer.

Activity	proto	src port	dst port	up pkts	down pkts	up bytes	down bytes
Super-peer signaling	udp	33837	26137	2	2	71	29
	tcp	51236	26137	161	97	19 k	9 k
Normal p2p signaling	udp	57046	33837	1	1	31	123
		33837	11229	3	3	527	497
		33837	17983	1	4	22	5 k
File transfer	udp	13524	33837	243	247	6 k	123 k
Call	udp	33837	13524	3 k	4 k	493 k	484 k

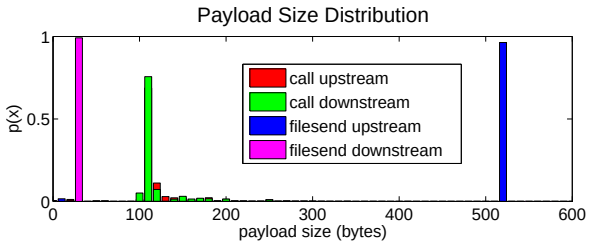

Figure 2: Voice Call vs File Transfer.

By analyzing ***host***-layer information, we found that port numbers are used more than once on a short period: this easily reveals which services the host is listening for (i.e. port 33837). Then, considering the ***service*** layer, we discover that the application uses port 33837 also for outgoing UDP connections. This also reveals both the UDP and TCP listening ports of peers on the other side. At ***biflow*** layer we see that signaling traffic is mostly composed by many short UDP biflows revealing a few different patterns: (I) some of them consist of only two packets (one per direction) of predictable size; (II) others present a single query packet and some response packets; (III) others present few packets in equal number for upstream and downstream directions with similar cumulative sizes. On the other side, file transfers present almost the same number of packets in both directions, but most bytes fall only in one of them. Finally, voice calls reveal a symmetric pattern in transferred data. At the ***packet*** layer, as shown in Fig. 2, the distribution of file transfer and voice-call payload sizes are significantly different. Combining the previous observations allows to identify Skype and its activities. For instance, the detection of many short UDP biflows related to the same service that show known patterns at packet level, allows to easily infer the Skype random port number. After that, it is straightforward to label all its communications in a port-based fashion (also for incoming TCP connections). Moreover, by inspecting the packet-size distribution of each biflow, it is also possible to discriminate between file transfers and voice calls.

As a final consideration, we can state that the proposed layered methodology permits to easily detect the presence, and to identify behavioral patterns, of a Skype client running on a host.

4. CONCLUSIONS AND FUTURE WORK

The methodology presented in this work is currently under study, to be refined and improved. We are currently testing it against other platforms (Facebook, Meebo, SecondLife, Cloud Platforms like GoogleDocs, ...) to demonstrate its validity. We also plan to implement it in TIE and to distribute it freely.

5. REFERENCES

[1] W. Li, M. Canini, A. W. Moore, and R. Bolla. Efficient application identification and the temporal and spatial stability of classification schema. *Elsevier Computer Networks.*, 53(6):790–809, 2009.

[2] A. Dainotti, W. de Donato, and A. Pescapé. TIE: A Community-Oriented Traffic Classification Platform. In *Proceedings of the First International Workshop on Traffic Monitoring and Analysis*, Berlin, Heidelberg, 2009.