Obtaining Application-based and Content-based Internet Traffic Statistics

Tomasz Bujlow and Jens Myrup Pedersen
Section for Networking and Security, Department of Electronic Systems
Aalborg University, DK-9220 Aalborg East, Denmark
{tbu, jens}@es.aau.dk

***Abstract*—Understanding Internet traffic is crucial in order to facilitate the academic research and practical network engineering, e.g. when doing traffic classification, prioritization of traffic, creating realistic scenarios and models for Internet traffic development etc. In this paper, we demonstrate how the Volunteer-Based System for Research on the Internet, developed at Aalborg University, is capable of providing detailed statistics of Internet usage. Since an increasing amount of HTTP traffic has been observed during the last few years, the system also supports creating statistics of different kinds of HTTP traffic, like audio, video, file transfers, etc. All statistics can be obtained for individual users of the system, for groups of users, or for all users altogether. This paper presents results with real data collected from a limited number of real users over six months. We demonstrate that the system can be useful for studying the characteristics of computer network traffic in application-oriented or content-type-oriented way, and is now ready for a larger-scale implementation. The paper is concluded with a discussion about various applications of the system and the possibilities of further enhancements.**

***Index Terms*—Internet traffic, traffic classification, computer networks, per-application statistics, per-content-type statistics**

I. Introduction

Monitoring traffic in computer networks and understanding the behavior of network applications is a very important challenge for both Internet Service Providers (ISPs) and scientists. ISPs focus on the business aspects of traffic monitoring, like improving the Quality of Service (QoS) in their networks. In order to setup the QoS rules in the network in a proper way, it is necessary to know what kind of traffic is flowing in the network, and how large amounts of traffic different applications account for. The knowledge of which applications are most frequently used in the network can be used by the ISPs to enhance the user experience by tuning some network parameters or setting up dedicated proxies or servers for particular applications or services. Users located in the same subnet can be compared and grouped according to their profile (like heavy user or interactive user), or distributed among the network to balance the load. Many ISPs have multiple connections to the external world, including many content deliverers. The knowledge of which connections is used most frequently can benefit in more accurate decisions from which provider the bandwidth should be bought. Finally, in many countries, the law obligates the ISPs to log all traffic, in order to be able to track down cybercrime, investigate terroristic attacks, etc. The knowledge of what the traffic is can benefit in saving storage space by logging only the important part of the traffic.

On the other hand, scientists use traffic monitoring to model traffic correctly and to create realistic scenarios of Internet usage. The models can be used for testing various options in designing networks before implementing them, examining the influence of a change in the current network design before applying it, or creating precise traffic classifiers.

There are many possibilities to obtain the relevant traffic statistics. Some data traces are available to the public (as Caida sets [1]), but most of them lack the detailed information about each packet (like payload, status of TCP flags), or about the structure of the flow (like inter-arrival times of the packets). Without the access to the real data, it would not be possible to conduct many interesting studies [2], [3]. There are, however, many possibilities to obtain the traces directly from the network by researchers. Unfortunately, this method has several drawbacks. First of all, these traces are obtained only in a few selected points, to which the researcher has access, so the traces are geographically limited. This concerns for example collecting data by Wireshark [4], or Cisco Netflow [5], which can provide some good statistics in the selected points of the network. Second, the obtained traces must be pre-classified according to the application-layer protocol, type of content carried by particular flows, etc. This task is not trivial, and it is a hard challenge to perform correct classification, especially when subject to real-time or near real-time requirements.

The simplest idea, widely used to pre-classify the traffic is using the application ports [6], [7]. Unfortunately, this fast method can be applied only to the applications or protocols, which use fixed port numbers. Nowadays, most traffic is generated by applications of Peer-to-Peer (P2P) nature, which operate on dynamic port numbers. Therefore, through port-based classification it is not possible to detect Bittorrent, or Skype [4], [8], [9].

The second commonly used solution to pre-classify data is Deep Packet Inspection (DPI). However, the name of this method can be misleading, since many DPI tools rely in fact on statistical parameters and they perform statistical classification to discover some applications. It causes some overlap and produces false positives and false negatives [10], [11]. Furthermore, DPI is quite slow and it require a lot of resources, especially processing power [4], [8]. Processing payloads of the users' data also raises privacy and

confidentiality concerns [4].

To avoid the issues described above, we developed at Aalborg University a tool called Volunteer-Based System (VBS). The most articulate advantages of the system is that by monitoring at the host machines, we are able to see the traffic exactly as it is generated at the source, and we are able to see which applications are opening the sockets, enabling creating accurate mappings between applications and traffic flows. Even with a relatively low number of users, we can obtain good understanding of how different applications behave with respect to traffic, but in order to obtain the data which can be used to describe Internet usage a substantial amount of users would be needed.

This open source tool is released under *GNU General Public License v3.0* and published as a SourceForge project [12]. Both Windows and Linux versions are available. VBS is designed to collect the traffic from numerous volunteers spread around the world and, therefore, with a sufficient number of volunteers the collected data can provide us with a good statistical base. The task of the Volunteer-Based System is to collect flows of Internet traffic data together with detailed information about each packet. The information about the application associated with each flow is taken from system sockets and appended to the flow description. Additionally, we collect the general information about the types of transferred HTTP contents, so we are able to distinguish various kinds of browser traffic. The system ideas were first described and initially implemented in [13], after which the design of our current Volunteer-Based System was described in [14]. Further improvements and refinements can be found in [15]. In parallel with our efforts [16] describes a Windows-based system which partially uses the same ideas of host based monitoring and accurate application informations. Our system was used to obtain various statistics useful for Machine Learning, Quality of Service Assessment and traffic analysis [17]–[19]. Our last paper [20] demonstrated how the system can be used for generating statistics at the flow level.

In this paper, we present the possibilities of the system for creating application-based or content-type-based statistics on the flow and on the packet level. It presents the results of a 6 months test study of the system, based on data from 4 users who joined at different times during this period. The main contribution is the demonstration of how the system can determine which packets are generated by which applications, and even further specify the kind of data (for example, if traffic generated by a web browser is web, audio or video traffic). The paper is organized as follows: First, in Section II, we describe how the data is collected by the Volunteer-Based System and how the statistics are extracted. In Section III, we present the results, and in Section IV, we conclude the paper and discuss the further work.

II. Collecting Data by Volunteer-Based System

This section presents the brief overview of VBS. We tried to highlight parts which are relevant for collecting network data and associating it with particular applications and HTTP content-types. For more details about the design and implementation of VBS, please refer to our previous paper [15].

The Volunteer-Based system is built using the client-server architecture. Clients are installed among machines belonging to volunteers, while the server is installed on the computer located at Aalborg University. Each client registers information about the data passing computer's network interfaces. Captured packets are grouped into flows. A flow is defined as a group of packets which have the same local and remote IP addresses, local and remote ports, and using the same transport layer protocol. For every flow the client registers: anonymized identifier of the client, start timestamp of the flow, anonymized local and remote IP addresses, local and remote ports, transport protocol, anonymized global IP address of the client, and name of the application associated with that flow. The name of the application is taken from the system sockets. For every packet, the client additionally registers: direction, size, state of all TCP flags (for TCP connections only), time in microseconds elapsed from the previous packet in the flow, and type of transmitted HTTP content. We do not inspect the payload - the type of the HTTP content is obtained from the HTTP header, which is present in the first packet carrying this specific content. One HTTP flow (for example a connection to a web server) can carry multiple files: HTML documents, JPEG images, CSS stylesheets, etc. Thanks to that ability implemented in our VBS, we are able to split the flow and separate particular HTTP contents. The data collected by VBS are stored in a local file and periodically sent to the server. The task of the server is to receive the data from clients and to store them into the MySQL database.

The purpose of this study was to demonstrate the usage, so we focused on obtaining and presenting the data from a limited number of users. In future, we plan to make more wide-scale experiments. The statistics used in this paper were obtained from 4 users during the period from January to May 2012. However, the clients join VBS at different time points. The four users can be described as follows:

- User 1 - Private user in Denmark, joined the system on December 28, 2011
- User 2 - Private user in Poland, joined the system on December 28, 2011
- User 3 - Private user in Poland, joined the system on December 31, 2011
- User 4 - Private user in Denmark, joined the system on April 24, 2012

Our system was designed not only to store the complete knowledge of users' traffic in the Aalborg University database, but also to provide numerous useful statistics. These statistics can be calculated altogether, or grouped on a per-user basis, per-application basis, per-content-type basis, or on a mix of these. Most of them can be also calculated on a per-flow basis, and, therefore, they can be a direct input to various classification and clustering Machine Learning Algorithms. The calculated statistics include (but they are not limited to):

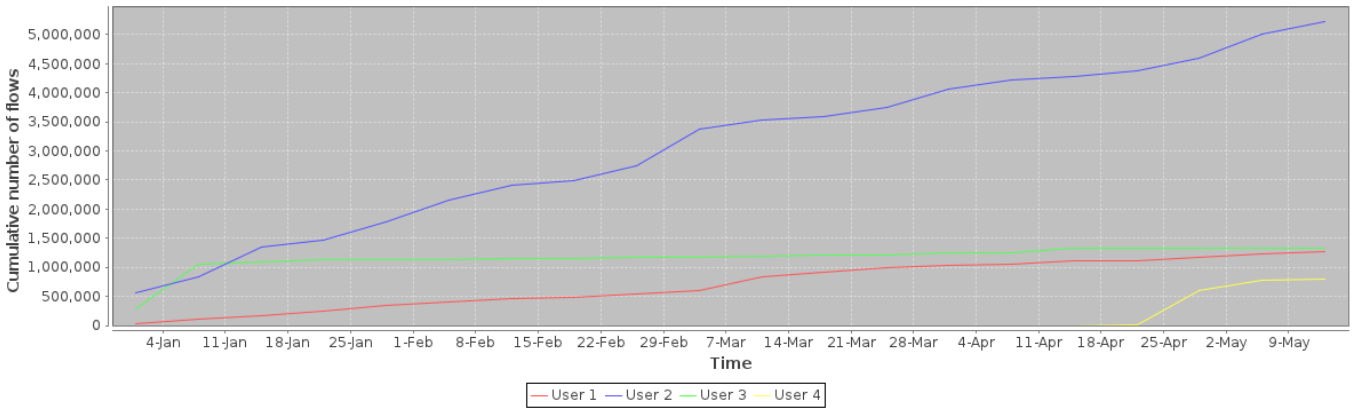

Figure 1. Cumulative number of flows belonging to different users over time.

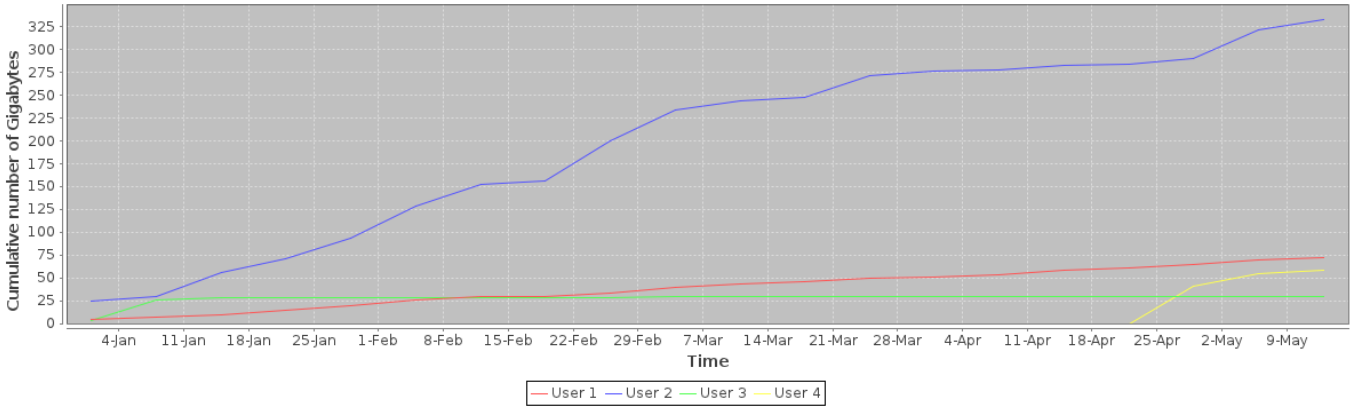

Figure 2. Cumulative amount of traffic belonging to different users over time.

- Number of flows
- Percent of all number of flows
- Average flow duration (in seconds)
- Average number of packets in flow
- Percent of inbound packets in flow
- Average inbound, outbound, and total packet size
- Minimum inbound, outbound, and total packet size
- Maximum inbound, outbound, and total packet size
- Median of inbound, outbound, and total packet size
- First quartile of inbound, outbound, and total packet size
- Third quartile of inbound, outbound, and total packet size
- Standard deviation of inbound, outbound, and total packet size
- Percent of inbound, outbound, and total packets which carry data
- Percent of data packets which are inbound
- Percent of inbound, outbound, and total data packets which are small (below 70 B)
- Percent of small data packets which are inbound
- Percent of inbound, outbound, and total data packets which are big (above 1320 B)
- Percent of big data packets which are inbound
- Percent of inbound, outbound, and total packets which have ACK flag
- Percent of packets with ACK flag which are inbound
- Percent of inbound, outbound, and total packets which have PSH flag
- Percent of packets with PSH flag which are inbound
- Amount of traffic (in Megabytes)
- Percent of traffic which is inbound
- Percent of traffic from all flows
- Number of TCP, UDP, and HTTP flows
- Amount of traffic (in Megabytes) carried by TCP, UDP, and HTTP flows

Due to limited length of this paper, we are not able to present and describe all the generated statistics. Instead, we decided to focus on a few per-application and per-content-type measurements, which we obtained for all users separately and altogether.

III. Results

A. Number of Flows vs Number of Bytes

The amount of traffic passing a network connection can be characterized using various metrics. The most common used are number of bytes and number of flows. They are dependent on each other, because the increasing number of network flows always increase the number of transferred bytes. However, flows can be short or long, and packets belonging to that flows

Table I
TOP 10 APPLICATIONS FOR ALL USERS.

Order no.	Application	Amount [MB]	% of all traffic	No. of flows	% of all flows	Avg. no. of packets in flow
1	uTorrent	348694	61	7151020	72	63
2	chrome	55675	9	599228	6	115
3	firefox	33657	5	381805	3	109
4	svchost	27994	4	118059	1	405
5	moc	20943	3	141557	1	179
6	java	18767	3	81379	0	280
7	libgcflashplay	12028	2	88	0	139567
8	libgcflashpla	8312	1	59	0	135731
9	Unknown	7395	1	1135040	11	11
10	SoftonicDownloader	7105	1	15035	0	509

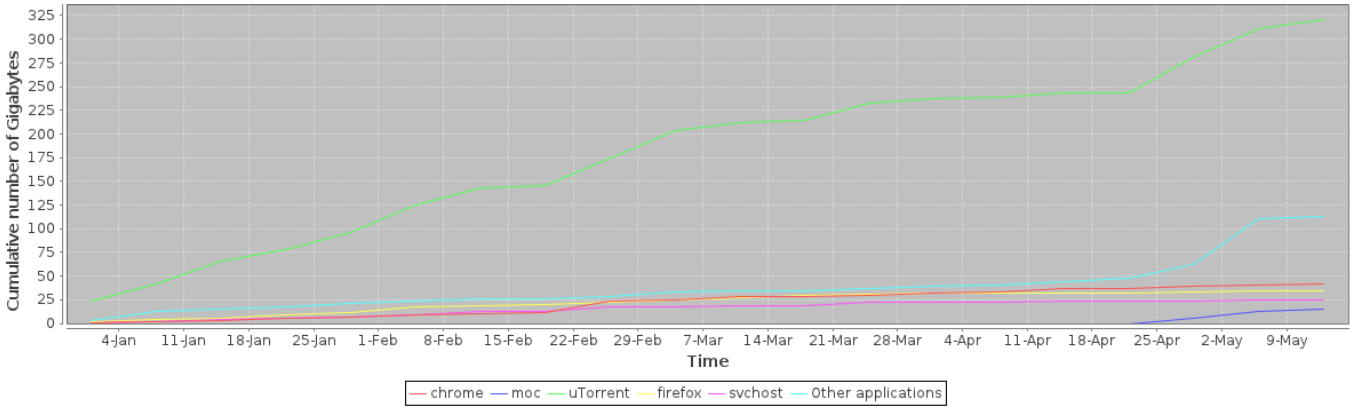

Figure 3. Cumulative amount of traffic generated by top 5 applications over time.

can have various lengths. Therefore, on different machines, the increase of number of flows have different impact on the increase of number of Bytes. The cumulative numbers of flows collected over the time from different machines are shown in Figure 1. Similarly, the cumulative numbers of bytes collected from the same clients over the same period of time are shown in Figure 2. The characteristics are quite similar - the difference concerns the client number 3. This user produces higher number of flows than users 1 and 4, but it generates the lowest traffic among all the users. It means that the user number 3 must use more interactive applications (producing smaller packets) than the other users, or use applications producing shorter flows. Based on that we can assume that this user is not a heavy downloader – file downloads usually use only a few flows, but each of them carries large amounts of data. Our suspicions will be proved in the next points, when we show the distribution of different applications among all the observed users.

B. Top 10 Applications

Analyzing the network traffic in the application-wise way is a very challenging task. Our Volunteer-Based System (VBS) is able to associate each flow with the application name, which is taken from the system sockets. This approach is quite straightforward, but unfortunately it also has one big drawback - the socket must be open for a sufficiently long time to allow VBS to notice it and to grab the application name. Consequently, a substantial number of short flows lack the associated application. During our research, 260 different process names accounting for 556.7 GB of data were identified, and for this study we just list the top 10 of them. To emphasize the influence of the flow length on the ability to obtain the application name, the average number of packets in flow is also included in these statistics.

The top application names for all users altogether are shown in Table I. We also include the information about the number of flows belonging to each application. The applications are ordered according to the amounts of transmitted data.

The obtained results show that:

- The average number of packets in flows without assigned application name is 11, comparing to 63–139567 in flows with the application name assigned. This confirms that our VBS is not good in providing application names for short flows. However, it is worth noticing that flows without assigned application name account only for 1 % of the whole traffic volume.
- A big number of flows does not mean big amount of data. Flows without assigned application name account for 11 % of of total number of flows (second position), but only for 1 % of the whole traffic volume (9. position).
- Applications having large number of packets in a flow (like *libgcflashplay* and *libgcflashplaya*, responsible for streaming video through web browser) can account for more traffic than applications having small number of

flows (like all applications belonging to the *Unknown* group together). Normally, it would not be surprising, but in this case the proportion of the number of flows belonging to *libgcflashplay* to the *Unknown* group is 1:12898.

The same statistics for individual users are shown in Tables II–V. The results also depict the reason for the phenomena described earlier in this paper. Most flows (91 %) belonging to user number 3 consist of 23 packets in average, comparing to 70–112 packets for other users. That is why we encounter more flows from user number 3 than from users 1 and 4, but we see lower amount of traffic.

The cumulative amount of traffic generated by top 5 applications for all users altogether are shown in Figure 3. It is clearly depicted that the amount of traffic generated by *uTorrent* is over 6 times bigger than the amount of traffic generated by the second biggest traffic provider in our chart (*chrome*), and 3 times bigger than the amount of traffic generated by all applications besides the top 5. Large amount of *uTorrent* traffic led us to study its behavior more carefully. The cumulative amounts of traffic generated by downloading and uploading files by *uTorrent* are shown in Figure 4 and Figure 5, respectively. The charts depict the very interesting characteristics of bittorrent traffic - downloading and uploading is realized simultaneously, so the download and upload curves have the same shapes. It is, however, worth noticing that the amount of downloaded traffic is around 7 times bigger than the amount of traffic uploaded by the clients. The next interesting observation is that user number 4 uploads almost the same amount of data as it downloads, so it is possible that he has a symmetric Internet connection.

C. Top 10 HTTP Content-Types

The previous subsection showed that besides the bittorrent traffic, web browsers account for the most of traffic transmitted in computer networks. This fact is not surprising since more and more services are becoming web-based, including web radio, web television, web applications, etc. Therefore, the knowledge of which application generated the traffic is not sufficient and we needed to perform the examination what the browser traffic is. Our Volunteer-Based System is able to provide us information about the *Content-Type* headers transmitted by the web server to the browser for each part of information received by the client. During our research, 191 different HTTP content-types accounting for 98.5 GB of data were identified, and for this study we just list the top 10 of them. Grouping such content-types into particular categories (like audio, video, binary data, etc) is outside the scope of this paper and it is a subject to further examinations.

The top HTTP content-types for all users altogether are shown in Table VI. The content types are ordered according to the amounts of transmitted data. Unlikely than when describing traffic generated by various applications (we were taking into account inbound as well as outbound traffic), we consider in this point only the inbound traffic. The reason is that only the inbound traffic is responsible for delivering the content to the clients. The outbound traffic while transmitting HTTP contents is very low and it consists of small packets containing acknowledgments and new parts requests. Table VII contains the comparison of the inbound and outbound characteristics of the traffic while downloading particular contents via HTTP.

The results show that the majority of HTTP traffic is generated by video and binary files downloaded by users. The web traffic, however, also occupies three places (*image/jpeg*, *text/html*, and *text/plain*) in the list of top 10 HTTP content-types. It is worth mentioning that these three content-types account for 52 % of the total number of transferred HTTP contents, but only for 9 % of the total number of transferred HTTP traffic, due to a low number of packets from which these contents consist of (4–9 in average).

The same statistics for individual users are shown in Tables VIII–XI. For each user, in the top 10 content-types we can find the ones characteristic for web browsing activities (*image/jpeg*, *text/html*, and *text/plain*) and the ones characteristic for video services, as *YouTube* (*video/x-flv*). The latter content-type is also commonly used by Video on Demand (VoD) applications, as *Ipla*, which are not web browsers, but they use HTTP to download video data to the user's computer. The other interesting dependency, which can be noticed based on these four tables is the inverse proportionality between the number of observed occurrences of the particular content-type and the average number of packets contained by the content. It means that most often we observe relatively short contents (as HTML files or web images), and larger ones are more rare (as movies).

The cumulative amount of traffic generated by the top 5 HTTP content-types for all users altogether are shown in Figure 6. It is clearly depicted that the amount of traffic generated by *video/x-flv* is around 2.5 times bigger than the amount of traffic generated by the second biggest traffic provider in our chart (*application/octet-stream*), and it also corresponds to the amount of traffic generated by all HTTP content-types besides the top 5.

D. Characterizing Application Traffic

With the data collected it is possible to characterize traffic from different applications by a large number of metrics. In this section, we will shortly demonstrate some of the interesting metrics, which can be used to characterize traffic based on the data collected throughout the study:

- Average packet sizes: inbound, outbound, and total
- Distribution of inbound and outbound packets
- Distribution of inbound and outbound packets carrying data

The results are presented in Table XII. Quite a few interesting observations can be made. While not surprising, it is interesting to observe that for *chrome* 60 % of the packets are inbound. If only packets carrying data are taken into account, this number increases to 71%. For *dropbox*, which was extensively analyzed on the flow and volume level in [21], it is interesting to note that while the number of inbound and

Table II
TOP 10 APPLICATIONS FOR USER 1.

Order no.	Application	Amount [MB]	% of all traffic	No. of flows	% of all flows	Avg. no. of packets in flow
1	firefox	29921	28	330734	21	112
2	chrome	27143	26	242937	15	142
3	libgcflashplay	12028	11	88	0	139567
4	libgcflashplaya	8312	8	59	0	135731
5	Unknown	5449	5	869134	56	9
6	http	4718	4	3104	0	1520
7	plugin-contain	2632	2	413	0	8058
8	iplalite	2618	2	473	0	5661
9	clwb3	2525	2	266	0	11300
10	filezilla	2249	2	62	0	38528

Table III
TOP 10 APPLICATIONS FOR USER 2.

Order no.	Application	Amount [MB]	% of all traffic	No. of flows	% of all flows	Avg. no. of packets in flow
1	uTorrent	295981	80	5379088	89	70
2	svchost	27757	7	115166	1	413
3	chrome	25703	6	272772	4	112
4	java	14746	3	42703	0	417
5	firefox	1660	0	5144	0	354
6	Unknown	1154	0	152489	2	17
7	skype	842	0	18452	0	307
8	thebat	339	0	1908	0	238
9	SoftwareUpdate	223	0	32	0	7327
10	dropbox	145	0	7106	0	74

Table IV
TOP 10 APPLICATIONS FOR USER 3.

Order no.	Application	Amount [MB]	% of all traffic	No. of flows	% of all flows	Avg. no. of packets in flow
1	uTorrent	19315	62	1267869	91	23
2	SoftonicDownloader	7105	22	15035	1	509
3	chrome	2819	9	83349	5	46
4	java	1524	4	2214	0	849
5	svchost	121	0	237	0	550
6	Unknown	66	0	24197	1	10
7	Pity	28	0	33	0	914
8	e-pity2011	20	0	12	0	1800
9	AcroRd32	1	0	15	0	94
10	AdobeARM	0	0	11	0	31

Table V
TOP 10 APPLICATIONS FOR USER 4.

Order no.	Application	Amount [MB]	% of all traffic	No. of flows	% of all flows	Avg. no. of packets in flow
1	uTorrent	33395	50	504063	52	78
2	moc	20943	31	141557	14	179
3	iexplore	3760	5	81306	8	60
4	java	2494	3	36459	3	86
5	firefox	2074	3	45927	4	59
6	vmnat	2015	3	2983	0	709
7	Unknown	722	1	89220	9	15
8	mantra	259	0	7746	0	49
9	javaw	160	0	158	0	1420
10	svchost	113	0	2656	0	51

Table VI
TOP 10 HTTP CONTENT-TYPES FOR ALL USERS.

Order no.	Content-type	Amount [MB]	% of all HTTP traffic	No. of contents	% of all	Avg. no. of packets
1	video/x-flv	35828	34	16238	0	1543
2	audio/mpeg	11884	11	1945	0	4355
3	application/octet-stream	8832	8	17688	0	351
4	application/x-msdos-program	7095	6	1673	0	2963
5	video/mp4	5987	5	5983	0	696
6	image/jpeg	5888	5	516090	18	9
7	application/x-debian-package	5119	4	2444	0	1447
8	application/zip	3278	3	309	0	7426
9	text/html	2398	2	618695	22	4
10	text/plain	2013	2	348826	12	6

Table VII
CHARACTERISTICS OF INBOUND AND OUTBOUND TRAFFIC FOR TOP 10 HTTP CONTENT-TYPES FOR ALL USERS.

Order no.	Content-type	Avg. inb. packet size [B]	Avg. outb. packet size [B]	% of inb. packets	% of inb. Bytes
1	video/x-flv	1499	51	64	98
2	audio/mpeg	1470	48	57	97
3	application/octet-stream	1474	46	64	98
4	application/x-msdos-program	1498	41	65	98
5	video/mp4	1244	104	58	94
6	image/jpeg	1496	47	66	98
7	application/x-debian-package	1516	48	67	98
8	application/zip	1496	41	65	98
9	text/html	880	226	53	81
10	text/plain	1032	149	63	92

Table VIII
TOP 10 HTTP CONTENT-TYPES FOR USER 1.

Order no.	Content-type	Amount [MB]	% of all HTTP traffic	No. of contents	% of all	Avg. no. of packets
1	video/x-flv	23757	45	6490	0	2554
2	audio/mpeg	6693	12	198	0	24380
3	video/mp4	3428	6	550	0	4316
4	application/x-debian-package	3319	6	202	0	11190
5	image/jpeg	3019	6	271485	21	9
6	application/octet-stream	2401	4	6636	0	261
7	text/html	1184	2	232431	18	5
8	text/plain	1156	2	168202	13	6
9	video/webm	807	1	30	0	18612
10	image/png	781	1	86470	6	8

Table IX
TOP 10 HTTP CONTENT-TYPES FOR USER 2.

Order no.	Content-type	Amount [MB]	% of all HTTP traffic	No. of contents	% of all	Avg. no. of packets
1	video/x-flv	5553	35	6210	0	632
2	video/mp4	2083	13	5311	0	276
3	application/octet-stream	1840	11	6777	0	194
4	image/jpeg	1473	9	111872	14	11
5	text/html	492	3	133701	17	5
6	text/plain	442	3	139826	18	4
7	application/rar	431	2	2	0	152136
8	image/png	429	2	43620	5	9
9	application/x-shockwave-flash	404	2	11304	1	27
10	application/x-javascript	298	2	39194	5	7

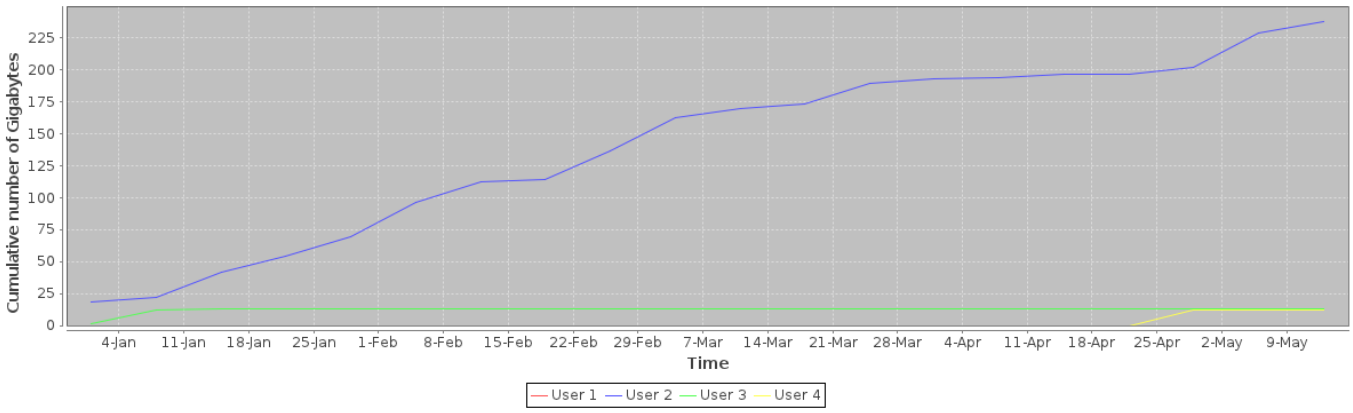

Figure 4. Cumulative amount of traffic downloaded by *uTorrent* over time.

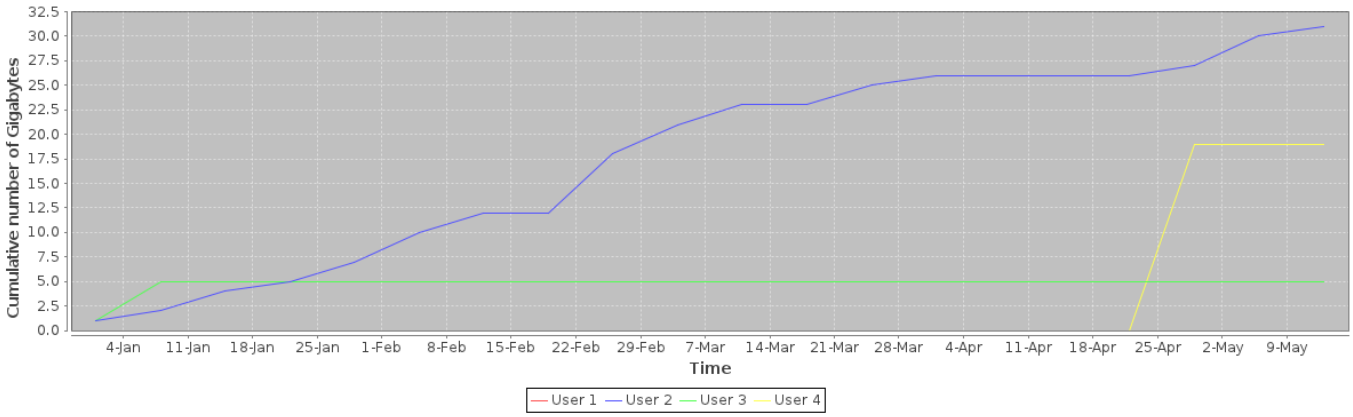

Figure 5. Cumulative amount of traffic uploaded by *uTorrent* over time.

Table X
TOP 10 HTTP CONTENT-TYPES FOR USER 3.

Order no.	Content-type	Amount [MB]	% of all HTTP traffic	No. of contents	% of all	Avg. no. of packets
1	application/x-msdos-program	6913	71	1440	1	3366
2	video/x-flv	1264	12	560	0	1604
3	image/jpeg	195	2	21811	15	9
4	application/x-shockwave-flash	173	1	3583	2	36
5	video/mp4	157	1	33	0	3373
6	image/png	148	1	21156	14	7
7	application/x-javascript	141	1	16095	11	8
8	application/x-compress	141	1	1	0	99186
9	application/octet-stream	101	1	313	0	229
10	text/html	87	0	26875	18	4

outbound packets are approximately the same, there is actually quite a large difference in the size of inbound and outbound packets. Looking into the more detailed figures for *dropbox*, it can actually be seen that 49 % of the outbound data packets are big (above 1320 B), while this is only so for 12 % of the inbound data packets.

IV. CONCLUSION AND DISCUSSION

In this paper, we have demonstrated how the Volunteer-Based System developed at Aalborg University can be used for generating useful statistics of Internet traffic usage – statistics which are useful academic as well as practical network engineering purposes. The system is based on monitoring traffic on the host, which has several advantages over traditional approaches for traffic monitoring - in particular, it is possible to obtain precise mappings between the applications and the traffic generated, which is a big help when training statistical classifiers. The paper demonstrated some of the statistics that can be obtained using the system, and examples of how they can be further processed to useful statistical information.

We focused in our studies on statistics calculated for various network applications, and presented both the overall statistics

Table XI
TOP 10 HTTP CONTENT-TYPES FOR USER 4.

Order no.	Content-type	Amount [MB]	% of all HTTP traffic	No. of contents	% of all	Avg. no. of packets
1	audio/mpeg	5027	20	925	0	3808
2	video/x-flv	5005	20	2978	0	1185
3	application/octet-stream	4463	17	3962	0	788
4	application/zip	3062	12	123	0	17419
5	application/x-debian-package	1797	7	2242	0	560
6	image/jpeg	1169	4	110922	18	9
7	application/x-gzip	646	3	37	0	12700
8	text/html	642	3	225688	37	3
9	text/plain	381	1	37874	6	9
10	video/mp4	316	1	89	0	2495

Table XII
CHARACTERIZING TRAFFIC GENERATED BY VARIOUS 5 APPLICATIONS FOR ALL USERS.

Application name	Average inb. packet size [B]	Average outb. packet size [B]	Average total packet size [B]	% of packets inbound	% of packets outbound	% of data packets inbound	% of data packets outbound
chrome	1314	130	842	60	40	71	29
dropbox	272	832	562	48	52	43	57
skype	207	178	193	51	49	51	49
uTorrent	1133	351	810	58	42	61	39
wget	1501	54	864	55	45	55	45

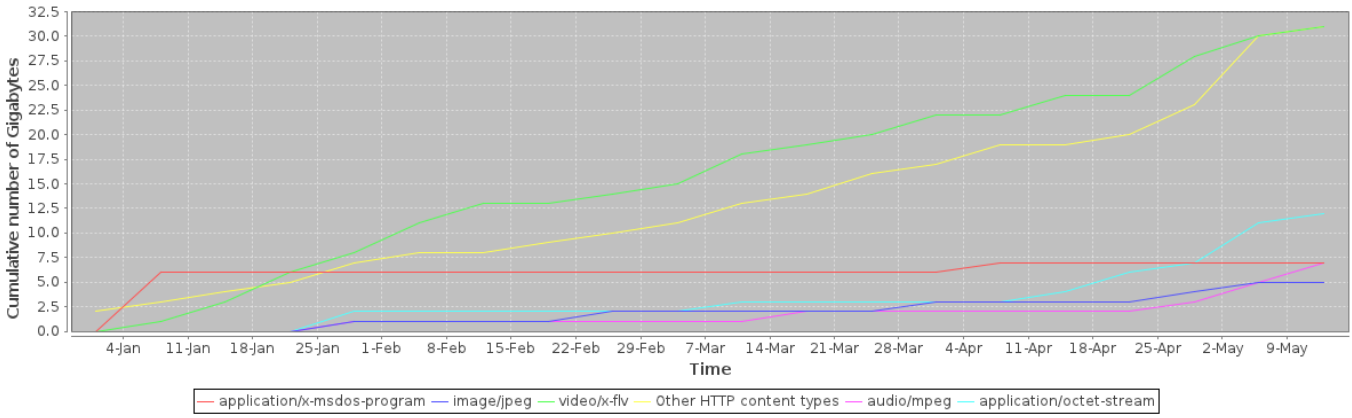

Figure 6. Cumulative amount of traffic generated by top 5 HTTP content-types.

and statistics that characterizes specific applications. Web browsers can carry today many different kinds of traffic, including interactive voice and video. We have demonstrated how we can use VBS to separate various types of HTTP traffic. The information gathered by the system can be used in many different ways: to create realistic models of computer networks, to provide accurate training data to Machine Learning Algorithms, to develop new and enhance existing networks. The current study has involved only a low number of volunteers in order to test the system prior to a large-scale implementation, and with the satisfactory results we are now ready to move on.

This paper is mainly intended as a demonstration of the system, and with the limited number of users the results do not represent the truth of neither application behavior nor distribution between applications. For the latter, it would be necessary to recruit not only a large number of volunteers, but also a set of volunteers representing the group that should be studied. For the former, a smaller number of users would be sufficient - as long as the group is large enough to ensure that different usages of the different applications are covered. That being said, we still believe that the results provide interesting indications of application behaviors for the most common applications such as Bittorrent and web traffic. It is important to keep in mind, though, that different user groups would still have different behaviors - for example, the use of e.g. web radios or web browsers could be different in different countries/regions depending on cultures, as well as between different user segments. If the system is to be used for training statistical classifiers, we would recommend that the data are collected from the same network as the classifier is later to be used in, in order to cope with these challenges.

Being aware that the amount of data is crucial, we highly encourage user groups and researchers to use the proposed system for collecting data and if possible sharing the anonymized traces with other researchers. Therefore, the system is based on open source code available from [12]. Other contributors would be welcome to set up their own servers for data collection, or to collaborate with the authors on the data collection.

Future research will focus on grouping the applications and the HTTP content-types into several sets, like voice, video, file transfer, interactive browsing, etc. This is not a trivial task, since grouping manually such large number of applications and content-types is not doable. Furthermore, the interactive connections (like interactive web browsing) should treat all the files in that connections as a whole, without splitting that into particular HTML documents, web images, stylesheets, etc. A kind of clustering algorithm can be used to partially automatize that process.

References

[1] CAIDA Internet Data – Passive Data Sources, 2012. [Online]. Available: http://www.caida.org/data/passive/.

[2] Chuck Fraleigh, Sue Moon, Bryan Lyles, Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, Ted Seely, and S. Christophe Diot. Packet-Level Traffic Measurements from the Sprint IP Backbone. *IEEE Network*, 17(6):6–16, November 2003. DOI: 10.1109/MNET.2003.1248656.

[3] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 90–102. ACM New York, Barcelona, Spain, August 2009. DOI: 10.1145/1644893.1644904.

[4] Jun Li, Shunyi Zhang, Yanqing Lu, and Junrong Yan. Real-time P2P traffic identification. In *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM 2008)*, pages 1–5. IEEE, New Orleans, Louisiana, USA, December 2008. DOI: 10.1109/GLOCOM.2008.ECP.475.

[5] Cisco IOS NetFlow, 2012. [Online]. Available: http://www.cisco.com/go/netflow.

[6] Riyad Alshammari and A. Nur Zincir-Heywood. Machine Learning based encrypted traffic classification: identifying SSH and Skype. In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009)*, pages 1–8. IEEE, Ottawa, Ontario, Canada, July 2009. DOI: 10.1109/CISDA.2009.5356534.

[7] Sven Ubik and Petr Žejdl. Evaluating application-layer classification using a Machine Learning technique over different high speed networks. In *Proceedings of the Fifth International Conference on Systems and Networks Communications (ICSNC)*, pages 387–391. IEEE, Nice, France, August 2010. DOI: 10.1109/ICSNC.2010.66.

[8] Ying Zhang, Hongbo Wang, and Shiduan Cheng. A Method for Real-Time Peer-to-Peer Traffic Classification Based on C4.5. In *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT)*, pages 1192–1195. IEEE, Nanjing, China, November 2010. DOI: 10.1109/ICCT.2010.5689126.

[9] Jing Cai, Zhibin Zhang, and Xinbo Song. An analysis of UDP traffic classification. In *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT)*, pages 116–119. IEEE, Nanjing, China, November 2010. DOI: 10.1109/ICCT.2010.5689203.

[10] Riyad Alshammari and A. Nur Zincir-Heywood. Unveiling Skype encrypted tunnels using GP. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, Barcelona, Spain, July 2010. DOI: 10.1109/CEC.2010.5586288.

[11] L7-filter Supported Protocols, 2012. [Online]. Available: http://l7-filter.sourceforge.net/protocols.

[12] Volunteer-Based System for Research on the Internet, 2012. [Online]. Available: http://vbsi.sourceforge.net/.

[13] Kartheepan Balachandran, Jacob Honoré Broberg, Kasper Revsbech, and Jens Myrup Pedersen. Volunteer-Based Distributed Traffic Data Collection System. In *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT 2010)*, volume 2, pages 1147–1152. IEEE, Phoenix Park, PyeongChang, Korea, February 2010.

[14] Tomasz Bujlow, Kartheepan Balachandran, Tahir Riaz, and Jens Myrup Pedersen. Volunteer-Based System for classification of traffic in computer networks. In *Proceedings of the 19th Telecommunications Forum TELFOR 2011*, pages 210–213. IEEE, Belgrade, Serbia, November 2011. DOI: 10.1109/TELFOR.2011.6143528.

[15] Tomasz Bujlow, Kartheepan Balachandran, Sara Ligaard Nørgaard Hald, Tahir Riaz, and Jens Myrup Pedersen. Volunteer-Based System for research on the Internet traffic. *TELFOR Journal*, 4(1):2–7, September 2012. Accessible: http://journal.telfor.rs/Published/Vol4No1/Vol4No1.aspx.

[16] Runyuan Sun, Bo Yang, Lizhi Peng, Zhenxiang Chen, Lei Zhang, and Shan Jing. Traffic Classification Using Probabilistic Neural Networks. In *Proceedings of the Sixth International Conference on Natural Computation (ICNC 2010)*, volume 4, pages 1914–1919. IEEE, Yantai, Shandong, China, August 2010. DOI: 10.1109/ICNC.2010.5584648.

[17] Tomasz Bujlow, Tahir Riaz, and Jens Myrup Pedersen. A method for classification of network traffic based on C5.0 Machine Learning Algorithm. In *Proceedings of ICNC'12: 2012 International Conference on Computing, Networking and Communications (ICNC): Workshop on Computing, Networking and Communications*, pages 244–248. IEEE, Maui, Hawaii, USA, February 2012. DOI: 10.1109/ICCNC.2012.6167418.

[18] Tomasz Bujlow, Tahir Riaz, and Jens Myrup Pedersen. A method for Assessing Quality of Service in Broadband Networks. In *Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT)*, pages 826–831. IEEE, Phoenix Park, PyeongChang, Korea, February 2012. Accessible: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6174795.

[19] Tomasz Bujlow, Tahir Riaz, and Jens Myrup Pedersen. Classification of HTTP traffic based on C5.0 Machine Learning Algorithm. In *Proceedings of the Fourth IEEE International Workshop on Performance Evaluation of Communications in Distributed Systems and Web-based Service Architectures (PEDISWESA 2012)*, pages 882–887. IEEE, Cappadocia, Turkey, July 2012. DOI: 10.1109/ISCC.2012.6249413.

[20] Jens Myrup Pedersen and Tomasz Bujlow. Obtaining Internet Flow Statistics by Volunteer-Based System. In *Fourth International Conference on Image Processing & Communications (IP&C 2012), Image Processing & Communications Challenges 4, AISC 184*, pages 261–268. Springer Berlin Heidelberg, Bydgoszcz, Poland, September 2012. DOI: 10.1007/978-3-642-32384-3_32.

[21] Idilio Drago, Marco Mellia, Maurizio M Munafo, Anna Sperotto, Ramin Sadre, and Aiko Pras. Inside Dropbox: Understanding Personal Cloud Storage Services. In *Proceedings of the 2012 ACM Internet Measurement Conference (IMC '12)*, pages 481–494. ACM, Boston, Massachusetts, USA, November 2012. DOI: 10.1145/2398776.2398827.