This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings

Today's Usenet Usage: NNTP Traffic Characterization

Juhoon Kim, Fabian Schneider, Bernhard Ager, Anja Feldmann
{jkim|fabian|bernhard|anja}@net.t-labs.tu-berlin.de
TU Berlin, Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, 10589 Berlin

***Abstract*—The finding by Maier et al. [1] that Network News Transport Protocol (NNTP) traffic is responsible for up to 5 % of residential network traffic inspires us to revisit today's Usenet usage. For this purpose we have developed an NNTP analyzer for the Bro network intrusion detection system. We find that NNTP is intensively used by a small fraction of the residential broadband lines that we study and that almost all traffic is to NNTP servers that require subscription for a monthly fee. The accessed content resembles what one might expect from file-sharing systems—archives and multimedia files. Accordingly, it appears that NNTP is used by some as a high performance alternative to traditional P2P file-sharing options such as eDonkey or BitTorrent.**

***Index Terms*—NNTP, Usenet, Internet usage, traffic characterisation, passive measurements**

I. Introduction

One constant in the Internet during the last 10 years has been its steady growth by more than 50 % each year [2], [3]. However, the traffic mix has undergone substantial changes. Initially, protocols such as FTP, SMTP, and NNTP were popular. Then, in about 1994, HTTP entered into the picture. Until 2000, P2P protocols such as Napster and Gnutella became popular but were later overtaken by eDonkey and BitTorrent. While there were still some hard-core NNTP users its usage declined during this time period and the Usenet was almost forgotten. In 2007 P2P contributed around 70 % of the Internet traffic [4]. Yet, several recent traffic studies report that there has been another substantial change in the application mix [1], [5]–[7]. HTTP is again responsible for more than 50 % of the traffic. Moreover, these studies find that NNTP is responsible for a significant fraction of the traffic. Most of the protocols from the early Internet age, e. g., SMTP and NNTP, are designed for exchanging text-based information. However, these days multimedia data such as video, audio, and pictures as well as RAR archives are among the dominant content [1], [5]–[7].

The Usenet evolved from its UUCP (RFC 976) architecture to a worldwide distributed Internet discussion system in which users read and post public messages to one or more categories, known as newsgroups, via NNTP. These messages are called articles or posts, and collectively are referred to as news. Usenet is similar to bulletin board systems and as such it is the precursor to various Internet forums. Today the traditional Usenet features are offered by forums and mailing lists and are integrated into blogs and online social networks (OSNs).

The Usenet is realized across a constantly changing set of servers that store and forward messages among each other. Since Usenet servers do not have unlimited disk space to hold every article that was ever posted, the oldest articles are deleted as new articles arrive. Usenet distinguishes between binary groups and text based groups since the storage and bandwidth requirements for binary groups are substantial. Due to its size a single binary message may squeeze out several hundreds of text-only postings and its download may impose significant bandwidth cost for the server provider. Therefore, NNTP administrators started to exclude binary groups on their publicly accessible news servers. As a consequence, NNTP became less desirable and eventually its usage declined as alternatives became available.

However, NNTP is again gaining popularity [1], [5], [7]. Given this revival it is important to understand its causes and examine its characteristics. Therefore, we developed an analyzer for the Bro [8] network intrusion detection system (NIDS). Our analyzer takes advantage of Bro's capabilities and utilizes dynamic protocol detection [9] and a specialized protocol semantic parser.

In this paper, we present observations from passive packet-level monitoring of more than 20,000 residential DSL lines from a major European ISP. This unique vantage point coupled with our NNTP protocol analyzer provides a broad view of its usage. We used Bro's online analysis capabilities to collect anonymized NNTP summaries which lasted for more than two weeks. In addition, we applied our analyzer to the traces also studied by Maier et al. [1].

To the best of our knowledge this is the first study of NNTP traffic since the advent of blogs and OSNs that provide NNTP like features in a shiny browser-based dress. Our initial expectation was that NNTP servers do not provide binary data. However, we find that most of the traffic (> 99 %) is binary content. We find that this is enabled by NNTP servers that require their users to pay a monthly fee and to authenticate themselves before using the server. Examples for such fee-based server providers are GigaNews or UseNext. Examining content-types of this binary traffic we find that archive formats as well as multimedia (AVI, MP3, and JPG) are dominating and are predominantly transfered with yEnc as binary-to-text encoding.

Furthermore, we note that such fee-based NNTP offers often provide customized NNTP clients (see for example Figure 1). Some of these clients take advantage of opening multiple simultaneous TCP connections and use globally unique article IDs instead of per news-group indexing. Moreover, the per

978-1-4244-6739-6/10/$26.00 ©2010 IEEE

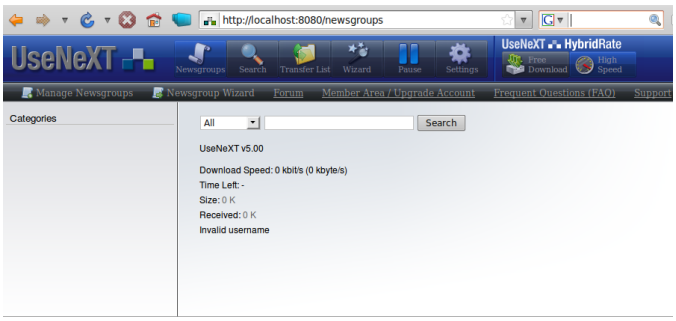

Fig. 1. Screen-shot of an NNTP client of a fee-based offer

connection throughput for NNTP is significantly larger for NNTP clients than for traditional P2P clients. Therefore, we conclude that NNTP's revival is due to the possibility of using NNTP as a high performance client/server-based alternative to P2P file-sharing, even though users have to pay a monthly fee.

The remainder of this paper is structured as follows: In Section II we give a short summary of NNTP. The design and implementation of the analyzer is discussed in Section III. After giving a short overview of our datasets in Section IV we present our results in Section V. We summarize in Section VI.

II. A REFRESHER ON NNTP

While NNTP is a well-known and well-established protocol its technical details might have faded from the readers memory. Therefore, we briefly review the basics of the protocol underlying the Usenet—NNTP—in this section. The Usenet is one of the oldest parts of the Internet. In fact, it predates the world wide web by more than 10 years. It was designed and is still being used as a system to exchange messages (called articles) between groups of users with similar interests—a functionality that today is also provided by, e. g., web forums.

The Usenet is structured in a (pseudo) hierarchy of groups, e. g., `comp.os.linux.networking` or `comp.os.minix`. Anyone with access to a news server can subscribe to any of the news groups and then read or post messages within the group. The news servers are usually hosted by ISPs or at universities[1]. They are connected to form an overlay network, which is used to replicate articles between servers so that everyone has access to all articles not just those posted to the local news server. However, there is one limitation: It is the decision of the administrator of a news server if a particular news group is hosted on his server, e. g., nowadays most news servers typically do not host the majority of the binary groups in order to avoid the risk of excessive bandwidth usage.

While NNTP is used for both client-server as well as server-server communication we in the remainder of this paper focus on client-server communication since our focus is on NNTP usage by residential users. The Network News Transfer Protocol (RFC 3977) and its message formats (RFCs 5536, 5537, until Nov 2009: RFC 1036) are very similar to SMTP. In fact, many of the article header fields are identical to the email message header fields, e. g., `From` and `Subject`. In addition, there are NNTP specific headers, for example the `Message-ID` header, which contains a unique identifier for an article valid on all news servers.

The IANA assigned ports 119/tcp for NNTP and 563/tcp for SSL-encapsulated NNTP (NNTPS) are the default ports for communication between an NNTP client and server. The dialog starts when the client contacts the server. The server answers with a greeting message. Then the client can issue its commands. The server replies to each client command with a three digit status code, a status message, and an optional data block in "multi-line" encoding which contains, e. g., the requested article. Likewise, a client can send a "multi-line" data block to the server, e. g., to post an article. At the beginning of the connection a news server may require authorization before the client can issue any commands. Overall, we identify 33 different NNTP commands that are either specified in one of the RFCs or offered by popular servers such as INN. Examples include selecting a group (`GROUP`), listing the articles within a group (`LISTGROUP` or (`X`)`OVER`), and fetching (`ARTICLE`, `BODY` and `HEAD`) or sending (`POST`) articles.

The article itself is composed of a newline-separated list of headers and an article body separated by a double newline[2]. Within NNTP everything is encoded as a "multi-line" data-block, including article bodies, or command results. This "multi-line" data-block format of NNTP imposes several restrictions on the content. For example it cannot contain NUL characters, it has a limited line length, and the period character (full stop) at the beginning of a line has to be escaped. Therefore, it is impossible to transfer binary data without encoding. Popular encodings are yEnc, UUencode, MIME, base64, BinHex, Quoted-Printable and XXEncode.

Note that MIME has just recently been standardized (RFCs 5536 and 5537) as transfer encoding for the Usenet while yEnc and UUencode are well established. UUencode has originated on System V and has traditionally been used to transfer email messages and Usenet articles before the corresponding TCP based protocols were in use. With Usenet in mind yEncode has been designed to minimize the overhead imposed by alternative encoding schemes. The idea is to only encode characters if it is absolutely required to adhere to the message format standard. The name yEncode is actually a wordplay: "Why encode?". While it is in principle possible that other binary encodings may be in use and may not be identified their traffic volume in our data sets is minimal.

III. METHODOLOGY

For our analysis we use the Bro NIDS [8] as network protocol analysis tool. Bro features TCP stream re-assembly, a scripting language making it easy to manage state and to generate reports, and BinPAC, a protocol parser generator [10]. Moreover, Bro supports dynamic protocol detection (DPD) [9].

[1]Users whose ISP does not offer access to a news server can subscribe for a small fee to independent servers, e. g., http://news.individual.net/

[2]A newline in this context is the two character string `\r\n`.

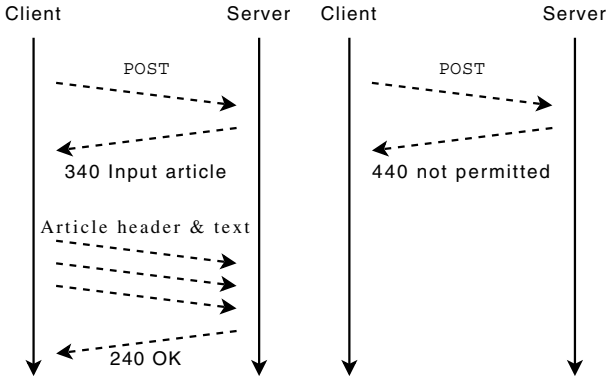

Fig. 2. Example scenarios of a `POST` transaction: Multi-line data is transmitted only when `POST` request is granted by the server (left).

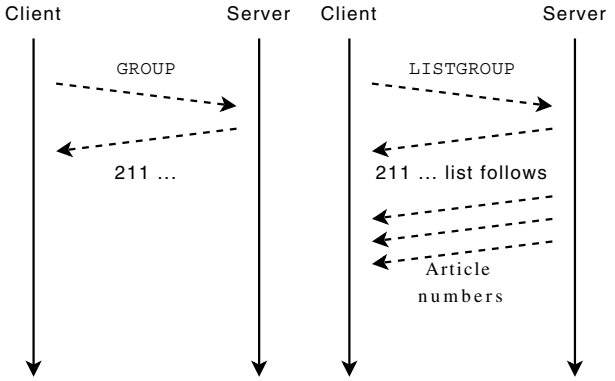

Fig. 3. Example scenarios of a 211 response code: Multi-line data is transmitted only when a `LISTGROUP` request was sent (right).

Our Bro analyzer for NNTP is also based on DPD. For each connection, the system first identifies potential protocols in use and then activates appropriate analyzers to verify the decision and extract higher-level semantics. As such the traffic classification is based on protocol analysis rather than port numbers and/or signature matching and we can detect NNTP usage on non-NNTP ports. However, we identified only a small fraction of NNTP traffic on non-NNTP ports. In addition, the protocol analysis enables us to understand which features/commands of NNTP are used. Moreover we can identify non-RFC conformant use of the NNTP protocol.

For detecting NNTP connections we use a DPD signature that checks if the server side of the connection contains an NNTP welcome banner and if the client side uses one of the 33 NNTP commands[3]. After a connection is identified as candidate for NNTP an analyzer written in BinPAC is attached to the connection and generates an event for every detected request command, reply message, and client and server side data sections (multi-line).

Note, that because NNTP is stateful, as opposed to, e. g., HTTP, our analyzer also needs to keep state per instance. Reasons for this necessity include client-side data transfers (e. g., via `POST`, `IHAVE`, or `XREPLIC` commands) or replies with or without multi-line data blocks depending on the request. In Figure 2, we show that if the client wants to post a multi-line article it first sends a single-line query, waits for a positive response from the server, and only then sends a multi-line data block. In Figure 3, we illustrate different reponse-types for status code 211 depending on the command issued by the client. Therefore, the analyzer needs to remember the last command for every connection, producing state that needs to be managed.

NNTP uses different encoding types to transfer binary content. To determine the encoding type we use the BinPAC analyzer itself instead of the scripting layer to lower CPU load. Thus, we add an event if a text-encoded binary payload is located to inform the scripting layer about encoding type and method. We distinguish between *(i)* yEncode (yEnc) [11], *(ii)* UUencode (RFC 976), *(iii)* MIME (RFC 2045), and *(iv)* non-binary content.

Articles with MIME attachments are identified by looking for and inspecting the `Content-Type` header. Content types and filenames are then extracted by parsing the MIME formatted body. UUencoded article bodies are detected by looking for an UUencode header in the article body. This header starts with the string '`begin␣`' followed by a three digit number denoting the UNIX permissions in octal notation, and a file name. The binary block ends with the string '`end`' on a single line. yEnc works similar to UUencode: the binary block starts with a string '`=ybegin␣`' followed up by parameters describing the length of each block, the size of the resulting file, and the file name on the same line. Each block ends with the string '`=yend␣`' followed by additional parameters.

Our methodology for determining the content-type of a binary encoded file is straightforward and relies either on the filename extension or on the content-type of the binary-to-text encoding. In future work we plan to extend the analyzer to decode the binary file and use libmagic to determine the file-type.

The analyzer produces an anonymized one line summary for every single or multi-line request or response including a time-stamp, a connection identifier, anonymized IP addresses of client and server, size of the action, and information about the observed action. This output is post-processed to generate one-line summaries of corresponding NNTP request and response pairs, called NNTP transactions in the remainder of the paper. In these one-line summaries we also report the news group in which the request was issued.

IV. Data Sets

We base our study on multiple sets of anonymized packet-level observations of residential DSL connections collected at aggregation points within a large European ISP. The monitor, using Endace monitoring cards, operates at the broadband access router connecting customers to the ISP's backbone. Our vantage point allows us to observe more than 20,000 DSL lines.

[3]To minimize the number of false positives we currently filter traffic on TCP port 25, mainly used by SMTP.

TABLE I
OVERVIEW OF ANONYMIZED PACKET TRACES AND SUMMARIES.

Name	Start date	Duration	Size	NNTP
NNTP-15d	Wed 05 Aug '09 3am	15¼ d	n/a	n/a
SEP08	Thu 18 Sep '08 4am	24 h	>4 TB	5 %
APR09	Wed 01 Apr '09 2am	24 h	>4 TB	2 %
AUG09	Fri 21 Aug '09 2am	48 h	>11 TB	2 %

Table I summarizes characteristics of the data sets, including their start, duration, size, and fraction of NNTP. We used Bro's online analysis capabilities to collect an anonymized trace summary which covers more than two weeks of NNTP traffic (NNTP-15d). Moreover, we use several anonymized one/two day packet traces collected over a period of 11 months (SEP08, APR09, AUG09). These are the same traces as studied by Maier et al. [1]. While we typically did not experience any packet loss, there are several multi-second periods (less than 5 minutes overall per packet trace) with no packets due to OS/file-system interactions.

All traces contain a noticeable fraction of NNTP traffic. Similar results have been observed at different times and at other locations of the same ISP. For the online trace summary we only capture NNTP traffic and therefore do not know the exact fraction. However, comparing the volume to other traces we presume that it corresponds to a similar fraction.

We omit analyzing NNTPS for two reasons: *(i)* due to the encrypted nature of NNTPS it is impossible to inspect the content and *(ii)* the amount of traffic on the respective port is small compared to the observed NNTP traffic. Indeed, we do not observe more than 0.3 % of total traffic on port 563, the standard SSL port of NNTP.

V. RESULTS

To understand the revival of NNTP we have to study how NNTP is used today. Accordingly, in this section, we start by studying the overall characteristics of our traces. Then we examine the volume distribution of NNTP transactions, the popularity of commands, encoding methods, content types, news groups, and NNTP servers.

A. NNTP Characteristics

In our traces of residential Internet traffic 2 % (5 % for SEP08) is identified as NNTP. We observe roughly 150 users in each of the packet level traces (SEP08, APR09, AUG09) and roughly 300 users in NNTP-15d. We also observe that the typical NNTP client uses multiple TCP connections in parallel (1st quartile: 2–3, median: 4–6, 3rd quartile: 7–8) and that up to 12 different servers are contacted by a single client—maybe to balance the load across different servers. Note that less than 1 % of the users (150 out of 20000) causes more than 2 % of the traffic with NNTP only. In addition these users are also among the top users in terms of per-line traffic contribution. This is confirmed by Maier et al. [1]. They also report that P2P use and NNTP use is unlikely to be observed at the same line.

Furthermore, we find that roughly 99 % (only 96 % in AUG09) of the transmitted volume is due to binary transfers. Yet, we notice that the fraction of binary queries is rising. The fraction is 55 % for SEP08 whereas it is roughly 90 % for the other traces (APR09, AUG09, NNTP-15d). Moreover, the average duration of observed NNTP connections[4] is 24 minutes for SEP08 while the average connection duration of all 2009 traces is 6 or 9 minutes. In order to understand the origin of these differences we investigate SEP08 in more detail. We find that there are three DSL lines in this trace that exhibit strange behaviors:

- One line is responsible for roughly 30 % of the NNTP volume in SEP08 (mainly via `BODY` commands). In all other traces the top contributing line is responsible for at most 10 % of the volume.
- Another line is responsible for more than 90 % of the `GROUP` and `STAT` commands in SEP08.
- A third line interacts with a server such that after normal NNTP activities the server sends tons of single-line responses with an unknown status code (not specified in the RFCs).

These three lines are responsible for roughly 48 % of the NNTP transactions and thus we also report results for SEP08 without these 3 lines, labeled as "SEP08-w3l". For example, we find that 79 % instead of 55 % are binary downloads for SEP08-w3l. Since, SEP08 also contains some regular news reading activity the results for SEP08-w3l are closer to those of the other traces but still differ. For privacy reasons, we do not investigate these 3 lines in detail.

B. Distribution of Transaction Volumes

Recall, that an NNTP transaction consists of an NNTP request and its corresponding NNTP response. Figure 4 depicts the cumulative distribution of the transaction volumes of each of the traces. This plot highlights several specifics of NNTP traffic. For instance, a single-lined message must not exceed 512 bytes. This results in the step around 100 bytes. The most frequent transaction sizes are 252 kB and the median transaction size 387 kB (see helper line in plot). They seem to correspond to the sizes of typical binary data blocks, which are parts of multi-part archives or multimedia files. We assume that these sizes are due to either the software used to partition the files or server restrictions on article length. Although we tried to confirm any of the explanations, we did not find evidence.

C. Popularity of NNTP Commands

Although we identified more than 30 possible NNTP commands and include all of them in our signature for detecting NNTP traffic, only 16 of them are observed in any of our traces. The overall frequencies of the top commands are listed in Table II. As expected, the most frequently issued command is `ARTICLE`. Interestingly, the `BODY` request is the next most popular and it is significantly more popular than `HEAD`. This

[4]For this statistic we only consider complete NNTP connections, i. e., those that contain the command `QUIT`.

TABLE II
FREQUENCY OF COMMANDS

Trace	reply only	ARTICLE	BODY	GROUP	STAT	HEAD	AUTHINFO	QUIT	MODE	XOVER
SEP08	15,9 %	43.6 %	15.7 %	14.1 %	6.4 %	2.1 %	1.0 %	0.1 %	0.1 %	<0.1 %
SEP08-w3l	-	82.6 %	8.1 %	0.9 %	-	4.1 %	1.8 %	0.2 %	<0.1 %	<0.1 %
APR09	-	83.2 %	10.3 %	1.1 %	-	<0.1 %	2.3 %	0.7 %	0.2 %	<0.1 %
AUG09	-	66.5 %	27.1 %	0.4 %	-	<0.1 %	2.0 %	0.2 %	1.2 %	<0.1 %
NNTP-15d	-	76.5 %	18.1 %	0.6 %	-	0.3 %	1.8 %	0.2 %	0.6 %	<0.1 %

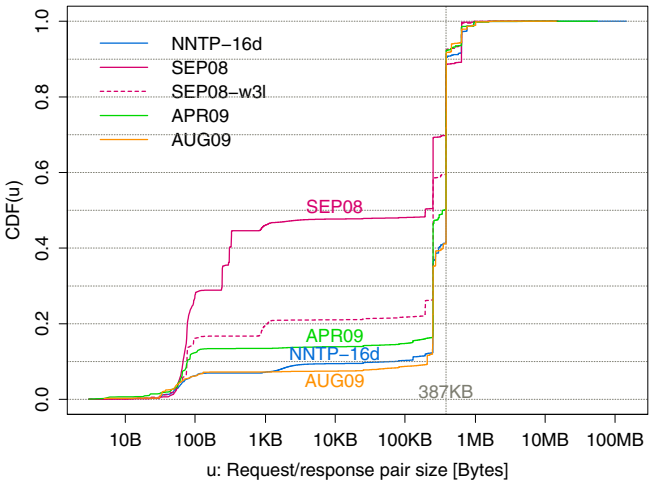

Fig. 4. Cumulative distribution function of transaction volumes

TABLE III
FREQUENCY OF BINARY-TO-TEXT ENCODING METHODS

Trace	non-binary	yEnc	UUEnc	MIME
SEP08	47.7 %	52.2 %	0.1 %	<0.1 %
SEP08-w3l	21.1 %	78.6 %	0.3 %	<0.1 %
APR09	14.7 %	84.8 %	0.5 %	<0.1 %
AUG09	9.8 %	89.9 %	0.3 %	<0.1 %
NNTP-15d	9.8 %	89.9 %	0.2 %	<0.1 %

TABLE IV
FREQUENCY OF BINARY FILE TYPES

Trace	archive /rar	archive /par2	video /avi	audio /mp3	image /jpg	other types
SEP08	84.30 %	2.57 %	5.19 %	1.46 %	0.40 %	6.08 %
APR09	84.13 %	1.73 %	7.08 %	2.36 %	2.07 %	2.63 %
AUG09	83.81 %	2.01 %	3.16 %	2.90 %	1.26 %	6.86 %
NNTP-15d	81.18 %	2.24 %	6.73 %	3.11 %	0.66 %	6.08 %

indicates that the users are not downloading the complete meta information about the bodies that they download. Note the effect of removing the 3 lines from SEP08.

In terms of volume only the commands ARTICLE and BODY are relevant and together contribute more than 99 %. NNTP offers two different identifiers for articles: *(i)* an article number relative to the newsgroup or *(ii)* a globally unique article identifier. The first kind of identifier requires the user to first select the corresponding newsgroup, e. g., via the GROUP command. The latter identifier can be used without entering a newsgroup. We find that the globally unique article identifiers dominate and may explain the small number of GROUP commands.

D. Popularity of Binary-to-text Encoding Methods

Given that most requests and almost all of the bytes are due to binary content we next explore which binary-to-text encodings are used. In all traces yEnc dominates with more than 99 % of the bytes being transfered (Table not shown). The frequency of requests for each encoding method is shown in Table III. Among the binary encodings yEnc is still dominant. We assume that yEnc's prevalence results from its significantly smaller encoding overhead.

E. Content-types of Binary Encoded Files

Given that almost all NNTP traffic is binary encoded we now explore which files are contained within these binary transfers, see Table IV. The most frequently transferred file format is archive/rar. RAR is a file compression software which is able to archive files and separate them into multiple smaller files. The file extension of separated files is three characters either 'rar', a 3-digit number or 'r' followed by a 2-digit number. Files for which the extension matches this condition have been assumed to be RAR files. We also observe parchive parity or index files (PAR2). These can be used to recover broken or unavailable data [12]. In addition, we see a significant fraction of multi-media formats. We note that in P2P file-sharing systems multimedia is more popular than archives according to ipoque [7].

F. Popularity of News Groups

Given our results so far we expect to see a large popularity of binary news groups. This is indeed the case. Most of the observed Usenet activities are in sub groups of alt.binaries. However, since most articles are requested via their globally unique ID we cannot identify which group they belong to. Moreover, for globally unique IDs it may be wrong to presume that an article belongs to the most recently selected group. Therefore, we do not present numbers in this paper.

G. Popularity of News Servers

Since binary news groups are usually not available on public NNTP servers we now examine which servers are contacted and apparently provide the desired binary content. To identify the server operator we use two approaches: *(i)* we parsed the

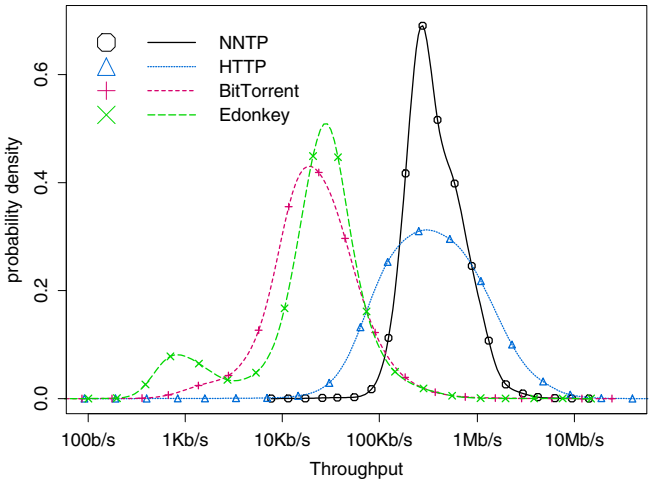

Fig. 5. Probability density function of achieved throughputs of flows $>$ 50 kBytes for different protocols in APR09.

welcome message line of the servers, *(ii)* we cooperated with the ISP to perform reverse lookups of the servers IP addresses. Both methods yield the same results: Most of the Usenet servers offering binary groups are commercial servers that require a monthly fee, i. e., GigaNews, UseNeXT, Alphaload, and Firstload. Further investigation reveals that there is even a reselling business for commercial NNTP server access.

Our investigation reveals that more than 93 % of the queries are directed to and more than 99 % of the volume of all NNTP traffic is exchanged with commercial Usenet servers. Browsing the web pages of these commercial Usenet server operators and taking advantage of some free trial offers we find that such offers come along with a special NNTP client. Clients for Unix-like OSes, e. g., MacOS or Linux, are often realized as background processes that are controlled via a Web browser by using machine local communication. These clients also provide a search engine for NNTP articles. This is an additional feature that NNTP does not provide by default. It is possible that such search queries are not using the NNTP protocol.

H. Throughput of NNTP

Next, we pose the question why people are willing to pay a monthly fee for something that is likely to also be freely available, e. g., via P2P file-sharing services. Therefore, we investigate the achieved throughput of NNTP flows and compare these to BitTorrent, eDonkey, and HTTP flows. Figure 5 shows a probability density function of the logarithm[5] of the achieved throughput for APR09. We can clearly see that NNTP outperforms the P2P-based systems. Given that both P2P-based systems and NNTP use multiple connections in parallel we assume that the overall time to download the same amount of data is an order of magnitude shorter for NNTP. This is basically due to the fact that NNTP servers usually have better Internet connectivity as compared to P2P users. For HTTP the achieved throughput is in the same order as for NNTP, although HTTP has a higher variability than NNTP. Thus, better performance may be the reason to either use fee-based NNTP servers or fee-based One-Click-Hosters (such as Rapidshare or MegaUpload) for file-sharing. One advantage for NNTP is the ability to search NNTP newsgroups for specific content. Moreover, NNTP features a protocol/operation inherent content replication and content distribution schemes.

[5]Coupled with a logarithmic scale on the x-axis, plotting the density of the logarithm of the data facilitates direct comparisons between different parts of the graphs based on the area under the curve.

VI. Summary

In this paper we presented our NNTP analyzer for the Bro NIDS and the analysis of how NNTP is used today. Our analysis is based on a live measurement lasting more than two weeks and several packet level traces. The most important observation is that more than 99 % of the volume are binary data transmissions. The distribution of transaction volumes shows that around 80 % of all transaction sizes are at two distinct peaks (252 kB and 387 kB). Furthermore, most traffic is exchanged with commercial servers that require a monthly fee from their customers. The achieved throughput of NNTP connections is at least an order of magnitude higher than P2P-systems like BitTorrent and eDonkey. These insights lead us to the conclusion that the Usenet is "misused" as a file sharing network.

In future work we plan to extend the analysis by using libmagic for content-type determination and including meta information contained in article headers (e. g., news groups).

References

[1] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband internet traffic," in *Proc. ACM Internet Measurement Conference*, 2009.
[2] A. M. Odlyzko, "Internet traffic growth: Sources and implications," http://www.dtc.umn.edu/mints/home.php, 2003.
[3] Global Internet Geography, "TeleGeography research," http://www.telegeography.com/product-info/gb/download/executive-summary.pdf, 2009.
[4] H. Schulze and K. Mochalski, "ipoque internet study 2007," http://www.ipoque.com/resources/internet-studies/ (need to register), 2007.
[5] Sandvine Inc., "2009 global broadband phenomena," http://www.sandvine.com/news/global_broadband_trends.asp, 2009.
[6] C. Labovitz, D. McPherson, and S. Iekel-Johnson, "NANOG 47: 2009 internet observatory report," http://www.nanog.org/meetings/nanog47/abstracts.php?pt=MTQ1MyZuYW5vZzQ3&nm=nanog47, 2009.
[7] H. Schulze and K. Mochalski, "ipoque internet study 2008/2009," http://www.ipoque.com/resources/internet-studies/ (need to register), 2009.
[8] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23–24, 1999.
[9] H. Dreger, A. Feldmann, M. Mai, V. Paxson, and R. Sommer, "Dynamic application-layer protocol analysis for network intrusion detection," in *Proc. Usenix Security Symp.*, 2006.
[10] R. Pang, V. Paxson, R. Sommer, and L. Peterson, "binpac: A yacc for writing application protocol parsers," in *Proc. ACM Internet Measurement Conference*, 2006, pp. 289–300.
[11] J. Helbing, "yEnc - Efficient encoding for Usenet and eMail," Project home page http://www.yenc.org/, 2003.
[12] "Parchive," http://parchive.sourceforge.net/.