Proceeding of the IEEE 28th
Canadian Conference on Electrical and Computer Engineering
Halifax, Canada, May 3-6, 2015

Deduping the Internet: An Email Case Study

Carey Williamson

Abstract— **Much of the traffic that traverses the Internet each day is redundant. That is, some or all of the data content has been sent previously. From a technical viewpoint, this represents a waste of resources, in terms of network bandwidth, storage, and energy efficiency. This paper presents an initial feasibility study to assess the potential of data deduplication technologies to reduce Internet traffic. The case study focuses on electronic mail (email), using an email dataset collected over the past 8 years. The results from this longitudinal study suggest that the size, complexity, and redundancy of email messages have all increased over this time duration, as has the complexity of the email delivery infrastructure. The results indicate that bandwidth savings of 30-45% are possible using existing redundant traffic elimination techniques on email messages.**

I. INTRODUCTION

The volume of Internet traffic continues to grow in an unabated fashion. In the early days of the Internet, the "big three" network applications were remote login, file transfer, and electronic mail (email). Through the decades, many other prominent network applications have appeared, including the Web, peer-to-peer file sharing, Voice over IP, video streaming, and social networking. This growth has been enabled by improved network access technologies, Internet penetration in more countries, additional users, and highly popular network applications.

Much of the traffic sent on the Internet is redundant, however. One of the reasons is related to Zipf's Law [1]. That is, there are highly popular Web sites and highly popular videos, which lead to skewed request frequencies, and many downloads of the same content. For this reason, the use of Content Delivery Networks (CDNs) and Web object caching are prevalent on the modern Internet [2]. These approaches are effective at detecting repeated references to the same object (e.g., based on URL), but are not as effective for aliased objects (i.e., different URLs that map to the same object) or slightly modified versions of objects. For the latter case of modified objects, delta-encoding techniques can be effective in reducing data transfer volume [3].

In storage systems, redundancy is typically addressed using data deduplication (deduping) techniques [4]. These algorithms identify blocks of data with identical content, store them once, and use compact and efficient fingerprinting techniques to identify and refer to them from elsewhere [5], [6].

On the Internet, similar techniques have also been applied. These techniques include redundant traffic elimination (RTE) [7], WAN optimization [8], delta-encoding [9], [10], and micro-caching of Web pages [11]. These techniques work by identifying chunks of repeated data, fingerprinting them, and caching them. Communicating entities can then transfer meta-data about the chunks, rather than sending the chunks more than once. RTE encoding records the locations and sizes of the repeated chunks, while decoding restores the original file content. WAN optimization typically reduces network traffic volume by 20-30% [12], though for Web servers savings as high as 65% have been observed [7]. These savings in network traffic are particularly important when operating over low-bandwidth networks, mobile cellular networks, or environments with constrained or transient network access [13], [14], [15].

In this paper, the primary goal is to consider the feasibility of applying data deduplication to Internet traffic. As an initial case study, the focus is on email traffic, using an empirical email dataset collected at the University of Calgary (with user consent). The paper focuses on characterization of the structural properties of this email corpus (e.g., size, complexity, and content type of emails) as well as the complexities of the email delivery infrastructure. Opportunities for data deduplication and redundancy elimination are identified. The results from this study indicate that 30-45% savings in bandwidth, storage, and communication costs are possible for email traffic, if current RTE techniques are incorporated into the SMTP delivery infrastructure.

The rest of this paper is organized as follows. Section II presents background information on Internet email and RTE. Section III presents a workload characterization study highlighting the structural properties of the email data set, and how they have changed over time. Section IV studies the inherent redundancy in email traffic, and quantifies the potential savings with data deduplication techniques. Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

A. Internet Email

The primary protocol used to exchange electronic mail on the Internet is SMTP (Simple Mail Transfer Protocol) [16]. SMTP is an application-layer protocol that operates in a peer-to-peer fashion between mail exchange (MX) servers. The originator of an email message typically uses SMTP to upload the message to the local SMTP server, which then forwards the message through one or more SMTP servers until it reaches its intended destination. The recipient then uses an email access protocol (e.g., IMAP, POP, or HTTP) to obtain and view the email using their preferred mail reading client (e.g., Web browser, Thunderbird, pine).

This work was supported in part by an NSERC Discovery Grant

C. Williamson is with the Department of Computer Science, University of Calgary, 2500 University Drive NW, Calgary, AB carey@cpsc.ucalgary.ca

978-1-4799-5829-0/15/$31.00 ©2015 IEEE

Email messages consist of a header, a body, and optional attachments. The header provides meta-data about a message, such as the sender, the recipient, the subject of the message, and the date it was sent. Optional header fields may be added by the sending email client to indicate content type, size, and attachments, and further header fields are added by SMTP to record information about the routing and timing along the delivery path to the recipient. The body of the message contains the main content, which is typically a few kilobytes (KB) in size, and encoded as readable ASCII text. Email messages can contain optional attachments, such as images, PDF files, spreadsheets, or signature information. Within SMTP, attachments are typically encoded using MIME (Multipurpose Internet Mail Extensions) [17] to permit ASCII-format transfers. The encoding records the type and size of the attachment, providing information that the receiving email client can use to extract and view the attachment properly.

B. *Deduplication*

Data deduplication is a widely used technique in storage systems, particularly in large-scale enterprise systems [4], [18]. In general, deduping saves about an order of magnitude (i.e., a factor of 10) on the amount of disk space required to store files. Savings occur because many files are stored multiple times by different users, and many files that are stored are small variations of previous files (e.g., slightly edited versions of Word documents, updated versions of Web pages, or successive versions of source code files).

Conceptually, data deduplication is simple. In essence, it stores each unique data block only once, and uses pointers from other locations to record repeated references to the same data content. In storage systems, the data block size is typically 4 KB or 8 KB, and the contents of a data block are summarized compactly using a Rabin fingerprint [5] or an MD5 hash [19]. Since the fingerprint size (e.g., 128 bits) is much smaller than a block size, substantial savings arise whenever a duplicate block is detected. There is additional computational overhead required, of course, to encode and decode duplicate blocks, but the savings in storage space and data transfer time far outweigh these costs, making data deduplication a viable and valuable technology in modern commercial storage systems.

Similar principles have been applied to Internet traffic, a domain in which the technique is known as redundant traffic elimination (RTE) [12], [20], [21], [22]. Two main differences arise in this setting. First, the data block sizes are much smaller. Typical RTE solutions use data chunk sizes of 32 bytes or 64 bytes, with 64-bit fingerprints. These chunk sizes are much smaller than those used in storage systems, and are also much smaller than a typical Web object or IP packet size. For this reason, the term micro-caching is often used to refer to this technique. Second, the savings are typically much smaller. Commercial implementations of WAN optimization typically claim 20-30% savings in network bandwidth, though the savings vary with the type of content. For example, binary content for images and videos usually offers very little savings (0-5%), since these content types are already encoded in an efficient compressed format, while Web pages and text files offer more savings (30-50%) [23]. One study indicated that RTE at a Web server could reduce traffic by 65% [7].

The key components within an RTE implementation are content-defined chunking (CDC), an efficient fingerprinting mechanism, and a cache at each endpoint to record data chunks that are frequently reused. Implementations differ in the chunk selection algorithm, the fingerprinting method, and the cache management techniques. In our own prior work [23], [24], we recommended content-dependent sampling, using 64-byte chunks and 64-bit Rabin fingerprints. In particular, we used a dynamically adapted version of the SAMPLEBYTE algorithm [20] called DYNABYTE [24], with savings-based cache management [23]. Similar techniques are used for our analysis of email traffic redundancy in Section IV.

III. EMAIL TRAFFIC CHARACTERIZATION

This paper presents an initial feasibility study of the effectiveness of data deduplication techniques applied to Internet email traffic. This is an empirical study, which uses a large sample of email traffic collected at the University of Calgary over the past 8 years. Most of the email traffic is from the most recent 4-5 years, though some email samples date back as far as 2007. The email dataset was collected with permission, and is analyzed in aggregate without revealing any user identifiable information. This section presents a high-level workload characterization study of this email dataset, to identify longitudinal trends over time, and provide context for the redundancy analysis in Section IV.

Table I provides a statistical summary of the email dataset, while Figure 1 and Figure 2 show graphical summaries of selected email traffic characteristics. In general, the average size of email messages (header plus body) have increased with time, as has the average number of SMTP routing hops for message delivery. The proportion of messages with attachments seems to be fairly consistent (25-30%), as is the average number of attachments observed per message with attachments. However, the average size of each attachment observed has also increased over time.

These characteristics suggest good potential for RTE. In general, larger amounts of textual and HTML content have greater redundancy, and the increasing size of email messages provides this opportunity. A similar argument can be applied for the increasing size of attachments, but the effectiveness of RTE will depend on the content type of attachments as well. Last but not least, the increasing length of SMTP routing paths implies that byte savings, when present, are applicable across lengthy end-to-end paths, implying even greater byte-hop savings on the Internet. The next section assesses the redundancy of email messages in greater detail.

Figure 3 shows the results from analyzing the SMTP headers. This graph shows the Cumulative Distribution Function (CDF) for the number of SMTP routing hops on the

TABLE I

Workload Characterization Results for Empirical Email Dataset (2007-2014)

Item	2007	2008	2009	2010	2011	2012	2013	2014
Number of messages	24	58	53	1,196	1,558	3,583	5,607	5,903
External sender domain (%)	29.2%	63.8%	45.3%	28.6%	31.6%	26.2%	26.5%	33.2%
Mean message size (bytes)	5,631	7,632	7,102	8,215	10,493	10,269	10,101	10,253
HTML-based messages (%)	12.5%	24.1%	43.4%	32.3%	31.1%	27.0%	30.9%	28.0%
Messages with attachments (%)	16.7%	34.5%	35.8%	27.8%	33.4%	27.1%	24.6%	24.9%
Average number of attachments	1.50	2.20	1.53	1.80	2.02	1.64	1.68	1.68
Average size of each attachment (KB)	88	73	83	87	99	137	123	138
Average SMTP hops (all messages)	7.29	7.17	6.60	6.99	7.15	6.94	7.94	8.09
Average SMTP hops (external messages)	6.86	7.51	7.21	7.71	7.59	7.69	9.68	10.58

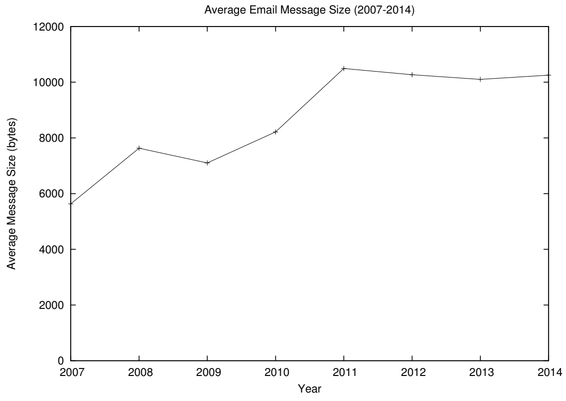

Fig. 1. Mean Size of Email Messages (2007-2014)

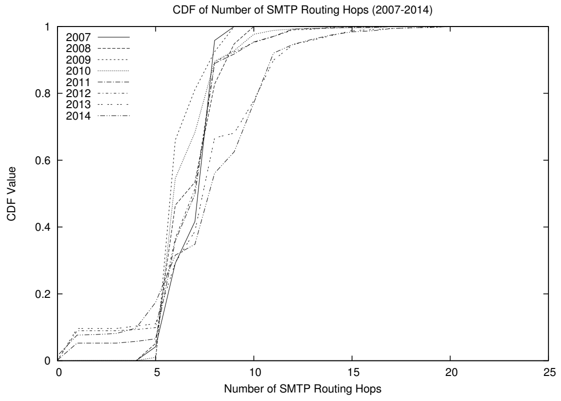

Fig. 3. CDF of SMTP Routing Hops (2007-2014)

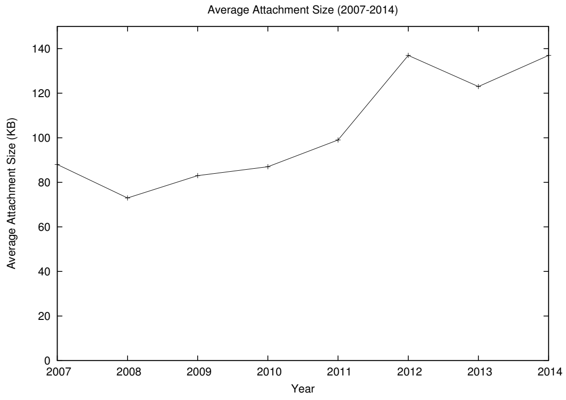

Fig. 2. Mean Size of Attachments (2007-2014)

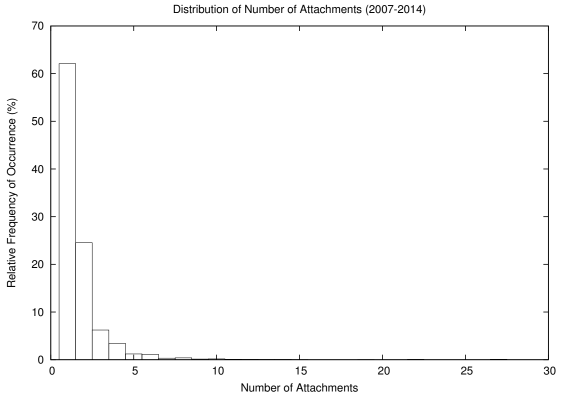

Fig. 4. Distribution of Number of Attachments (2007-2014)

email messages observed in each year of the data set. In 2007, there was a very tight distribution with at most 10 SMTP routing hops, while in 2014 there is a much wider spread to the distribution, with up to 25 SMTP routing hops observed. The growth in the number of SMTP routing hops is most pronounced for emails received from external sender domains, as illustrated in the last row of Table I. One reason for this increase is the recent addition of a third-party spam filtering service (Microsoft) for inbound emails from external senders. On average, this service adds about 4 routing hops to email message delivery.

Among the 26.3% of email messages that contain attachments, Figure 4 shows the distribution of the number of attachments observed. About 60% of these messages have a single attachment, while the maximum number of attachments observed was 27. The mean number of attachments observed on these messages is 1.7.

IV. Redundancy Analysis

In this section, we apply the principles of deduplication to our email dataset, using the Redundant Traffic Elimination (RTE) software that was described in Section II. In particular, we configure the software to use 32-byte chunks and 32-bit fingerprints, so there are 28 bytes of savings every time a redundant chunk is observed in the traffic.

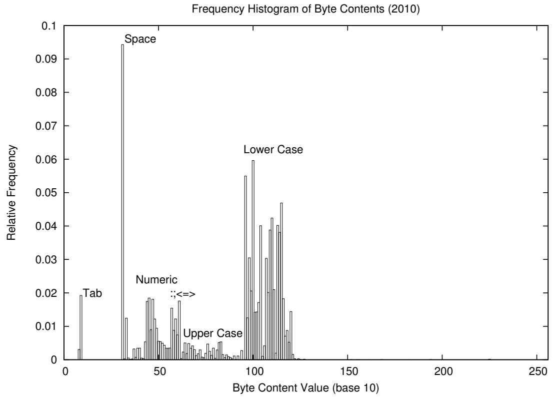

Fig. 5. Frequency Distribution of Byte Content Values (2010)

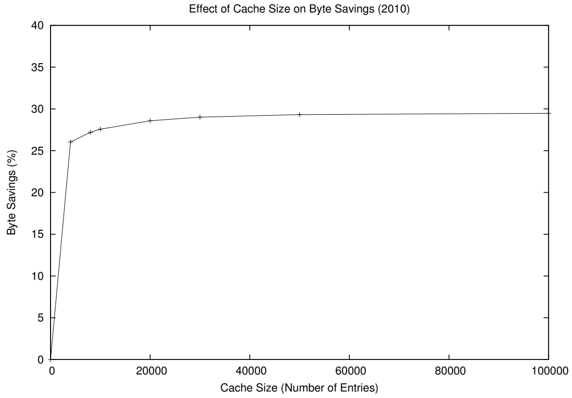

Fig. 6. Effect of Cache Size on Byte Savings (2010)

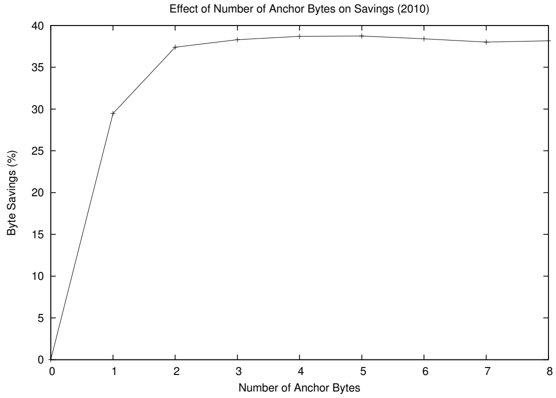

Fig. 7. Effect of Number of Anchor Bytes on Savings (2010)

TABLE II
FREQUENTLY OCCURRING CHUNKS IN EMAIL DATASET (2010)

Rank	Frequency	Data Chunk Content (in quotes)
1	1208	` Please contact IT Help Desk at ’
2	1094	` required 6.2, autolearn=disable’
3	1027	` class="MsoNormal"><span style="’
4	1021	` not spam, SpamAssassin (not cac’
5	832	` cms4.ucalgary.ca (Postfix) with’
6	775	` Found to be clean, Found to be ’
7	773	` UnhideWhenUsed="false" Name="’
8	495	` user=cwill bits=0) by cms4.uca’
9	495	` (Cyrus v2.2.12-Invoca-RPM-2.2.1’
10	484	` Carey Williamson <carey@cpsc.uc’
11	474	` class="MsoNormal"><o:p>&nbsp;</’
12	458	` style="margin-top: 0in; margin-’

We start by applying byte-frequency analysis on the contents of email messages to identify the most frequently occurring data bytes, and choose the most prevalent ones as our anchor bytes for chunk sampling. On our email message dataset, the most frequently occurring bytes are the printable ASCII characters ’ ’ (space), ’e’, ’a’, ’t’, ’o’, ’n’, and ’i’, since most of the messages contain English text. Other frequently occurring bytes are numeric values (for IP addresses and timestamps), ’:’ in date timestamps, and ’>’ in embedded email responses. Figure 5 shows the complete frequency distribution of byte values observed in the 2010 email dataset, as a representative example.

Figure 6 shows the effect of cache size on the byte savings of our RTE method, when only a single anchor byte (’e’) is used to trigger chunk selection for caching. With a single anchor byte, the savings plateau at about 28%, for any reasonable cache size. We use 100,000 entries (about 3 MB) as the cache size in our remaining experiments.

Figure 7 shows the effect of the number of anchor bytes on RTE savings. With a single anchor byte, the savings are 28%, as observed previously. As the number of anchor bytes is increased, additional chunks are added to the cache, creating more cache hits and more byte savings. However, the savings plateau near 38% with 4 anchor bytes, and decrease slightly beyond this point as greater contention for cache line entries occur. This experiment suggests that 4 anchor bytes are sufficient for extracting the maximum redundancy from our email message dataset.

Table II shows a dozen examples of frequently-occurring data chunks observed in the email dataset from 2010. These results are from using a single anchor byte ’ ’ (space). Among these examples, there are text strings from the spam filtering service (items 1, 2, 4, and 6), from HTML-based Web content (items 3, 7, 11, 12), and from other SMTP headers (items 5, 8, 9, and 10).

Figure 8 shows the complete distribution of the data chunk frequency on the 2010 email dataset for this example, while Figure 9 shows the resulting distribution when 5 anchor bytes are used for chunk sampling. Both graphs are plotted on a log-log scale, and exhibit a Zipf-like power-law distribution for the chunk popularity. The most popular chunks tend to be machine-generated ones, such as the SMTP headers and spam filtering headers illustrated previously. HTML content and user-generated email content are present throughout the rest of the distribution. With 5 anchor bytes on the 2010 email dataset, there were 283,442 chunks fingerprinted, of which 148,633 were new, and 134,809 were repeated occurrences. A total of 26,972 distinct chunks (about 10% of the total chunks fingerprinted) had repeated occurrences of the same data, although 15,581 of these (58%) were only reused once. However, the most frequently occurring

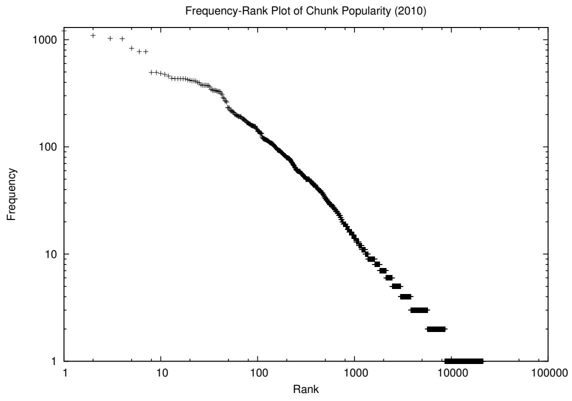

Fig. 8. Data Chunk Popularity Profile (2010, 1 anchor byte)

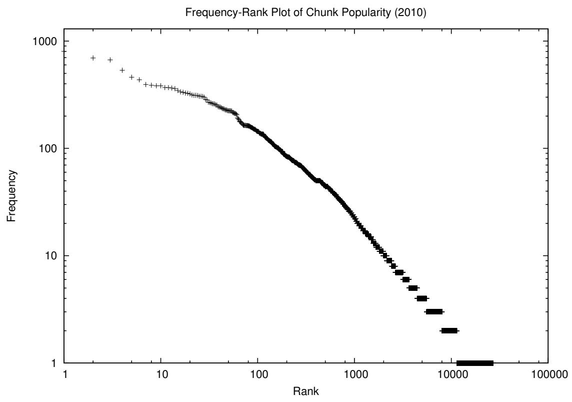

Fig. 9. Data Chunk Popularity Profile (2010, 5 anchor bytes)

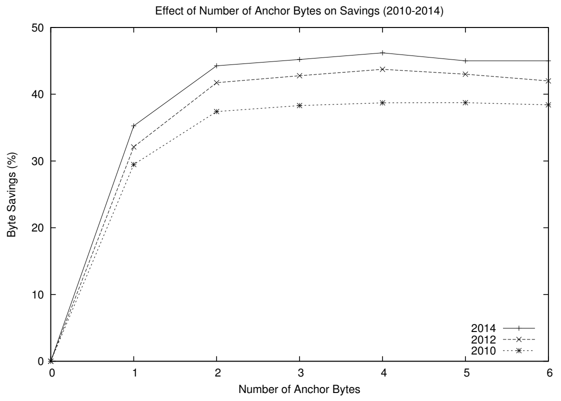

Fig. 10. Effect of Number of Anchor Bytes on Savings (2010-2014)

TABLE IV

ATTACHMENTS IN EMPIRICAL EMAIL DATASET

Type	Proportion	Avg Size (KB)
PDF	34.3%	230
Image (JPG, PNG, GIF, TIFF)	22.3%	39.2
Text/Plain	12.8%	1.8
Word	12.3%	112
Excel	5.8%	148
Text/HTML	3.5%	5.6
Octet-Stream	2.9%	128
Signature (PGP, PKCS, VCARD)	2.1%	15.6
Powerpoint	0.7%	410
Compressed (ZIP, GZIP)	0.4%	675
Other	2.8%	105
Total	100.0%	124

chunks had close[1] to 1000 occurrences. In aggregate, our RTE approach resulted in 38% byte savings on this dataset.

Table III shows the byte savings that result on the email dataset, with each year's worth of messages treated independently. As can be seen, byte savings of 30-45% are typical, indicating that our results are quite robust on this set of email traffic. Figure 10 shows a graphical summary of the results for three example years from the email dataset. The effectiveness of RTE seems to improve with the email traffic volume.

Table III also shows results for different data chunk sizes. With smaller data chunks, more cache hits occur, but the byte savings on each cache hit are smaller, since the fingerprint size is the same (32 bits). With larger data chunk sizes, the relative savings per cache hit are larger, but unfortunately the number of cache hits decreases more significantly. The end result is slightly lower byte savings for larger chunk sizes.

We next turn our attention to the attachments in the email messages. Table IV shows a breakdown of the relative frequency and average size of each type of attachment observed in the dataset. Adobe PDF (Portable Document Format) files are the most prevalent, representing about one-third of the attachments. On average, these are about 230 KB in size. Image attachments are the next most prevalent. This type includes JPG, PNG, GIF, and TIFF attachments, which collectively average 39 KB in size. The largest attachments observed are compressed files (ZIP and GZIP), which average 675 KB each.

To assess RTE savings, we conducted a manual analysis of a small sample of attachments. We chose 2 different versions of a 25-page Word document (approximately 110 KB in size), and 2 different PDF versions (approximately 450 KB in size) of the same document (based on the file names). These documents appear as a pair of attachments in two different email messages observed in late March 2014.

Table V shows the results of this analysis. In all experiments, we use 32-byte chunks, 32-bit fingerprints, 4 anchor bytes, and a cache size of 10,000 entries. For the Word document, the chosen anchor bytes are 0x0, 0x2, 0x4, and 0x64. There are minimal byte savings (1.50%) within the attachment itself the first time it is observed. However, on the second occurrence a day later, approximately 7% byte savings are possible using the RTE chunk cache. Similar observations apply for the PDF document, for which the chosen anchor bytes are 'e', '/', 't', and 'o'. On the first occurrence, negligible byte savings (0.5%) are possible with this attachment. However, the second occurrence shows 18% savings using the RTE chunk cache.

The primary observation from this study is that RTE

[1] Note that the use of multiple anchor bytes can reduce the effectiveness of other individual anchor bytes, since overlapping chunks are not permitted.

TABLE III

RTE BYTE SAVINGS FOR EMPIRICAL EMAIL DATASET (2007-2014)

Item	2007	2008	2009	2010	2011	2012	2013	2014
RTE Savings (16-byte chunks)	28.9%	27.4%	29.8%	38.0%	43.7%	40.0%	41.9%	43.0%
RTE Savings (32-byte chunks)	27.6%	25.2%	27.2%	38.8%	47.7%	43.7%	44.7%	46.2%
RTE Savings (48-byte chunks)	22.0%	19.1%	20.5%	33.1%	43.8%	39.9%	40.2%	42.3%
RTE Savings (64-byte chunks)	18.8%	15.5%	16.7%	29.2%	40.0%	36.4%	35.9%	37.9%

TABLE V

RTE RESULTS FOR SELECTED ATTACHMENTS

Attachment	Size (bytes)	RTE Savings (%)
Word v1 (March 27)	115,545	1.50%
Word v2 (March 28)	114,429	7.15%
PDF v1 (March 27)	458,764	0.54%
PDF v2 (March 28)	457,012	18.19%

savings are much greater on the email message content than on the attachments. However, savings are still possible on multiple versions of the same attachment.

V. CONCLUSIONS

This paper presents an initial feasibility study to assess the potential of data deduplication techniques to reduce Internet traffic. The case study focuses on electronic mail (email), using an email dataset collected over the past 8 years. The results from this longitudinal study suggest that the size, complexity, and redundancy of email messages have all increased over this time duration, as has the complexity of the email delivery infrastructure. The results indicate that bandwidth savings of 30-45% are possible using existing redundant traffic elimination techniques on email messages.

ACKNOWLEDGEMENTS

The author gratefully acknowledges Emir Halepovic and Keynan Pratt for developing the RTE encoding software implementations, and Caitlin Ryan for adding the delta-encoding support. Financial support for this work was provided in part by the Natural Sciences and Engineering Research Council (NSERC), via an NSERC Discovery Grant and an NSERC Undergraduate Summer Research Award (USRA).

REFERENCES

[1] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications", *Proceedings of IEEE INFOCOM*, New York, NY, March 1999.

[2] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", *IEEE/ACM Transactions on Networking*, Vol. 5, No. 5, pp. 631-645, October 1997.

[3] J. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy, "Potential Benefits of Delta Encoding adn Data Compression for HTTP", *Proceedings of ACM SIGCOMM*, Cannes, France, pp. 181-194, September 1997.

[4] U. Manber, "Finding Similar Files in a Large File System", *Proceedings of USENIX Winter Technical Conference*, San Francisco, CA, pp. 1-10, January 1994.

[5] M. Rabin, "Fingerprinting by Random Polynomials", Technical Report TR-CSE-03-01, Center for Research in Computing Technology, Harvard University, 1981.

[6] S. Schleimer, D. Wilkerson, and A. Aiken, "Winnowing: Local Algorithms for Document Fingerprinting", *Proceedings of ACM SIGMOD*, San Diego, CA, pp. 76-85, June 2003.

[7] N. Spring, and D. Wetherall, "A Protocol-independent Technique for Eliminating Redundant Network Traffic", *Proceedings of ACM SIGCOMM*, Stockholm, Sweden, pp. 87-95, August 2000.

[8] Cisco, "WAN Optimization and Application Acceleration", http://www.cisco.com/en/US/products/ps6870/.

[9] F. Douglis and A. Iyengar, "Application-specific Delta-encoding via Resemblance Detection", *Proceedings of USENIX Technical Conference*, San Antonio, TX, pp. 113-126, June 2003.

[10] W. Xia, H. Jiang, D. Feng, L. Tian, M. Fu, and Y. Zhou. "Ddelta: A Deduplication-Inspired Fast Delta Compression Approach", *Proceedings of IFIP Performance*, Turin, Italy, October 2014.

[11] X. Wang, A. Krishnamurthy, and D. Wetherall, "How Much Can We Micro-Cache Web Pages?" *Proceedings of ACM Internet Measurement Conference (IMC)*, Vancouver, BC, pp. 249-256, November 2014.

[12] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in Network Traffic: Findings and Implications", *Proceedings of ACM SIGMETRICS* Seattle, WA, pp. 37-48, June 2009.

[13] E. Halepovic, M. Ghaderi, and C. Williamson, "On the Performance of Redundant Traffic Elimination in WLANs", *Proceedings of ICC*, Ottawa, ON, June 2012.

[14] A. Muthitacharoen, B. Chen, and D. Mazieres, "A Low-bandwidth Network File System", *Proceedings of ACM SOSP*, Lake Louise, AB, Canada, pp. 174-187, October 2001.

[15] T. Suel, P. Noel, and D. Trendafilov, "Improved File Synchronization Techniques for Maintaining Large Replicated Collections over Slow Networks", *Proceedings of International Conference on Data Engineering (ICDE)*, Boston, MA, pp. 153-164, March/April 2004.

[16] J. Postel, "Simple Mail Transfer Protocol", Request for Comments (RFC) 821, August 1982.

[17] Internet Engineering Task Force, "Multipurpose Internet Mail Extensions", Request for Comments (RFC) 2045, 2046, and 2047, November 1996.

[18] P. Kulkarni, F. Douglis, J. Lavoie, and J. Tracey, "Redundancy Elimination within Large Collections of Files", *Proceedings of USENIX Technical Conference*, Boston, MA, pp. 59-72, June/July 2004.

[19] R. Rivest, "The MD5 Message-Digest Algorithm", Request for Comments (RFC) 1321, April 1992.

[20] B. Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese, "EndRE: An End-System Redundancy Elimination Service for Enterprises", *Proceedings of USENIX NSDI*, San Jose, CA, pp. 419-432, April 2010.

[21] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet Caches on Routers: The Implications of Universal Redundant Traffic Elimination", *Proceedings of ACM SIGCOMM*, Seattle, WA, pp. 219-230, August 2008.

[22] A. Anand, V. Sekar, and A. Akella, "SmartRE: An Architecture for Coordinated Network-wide Redundancy Elimination", *Proceedings of ACM SIGCOMM*, Barcelona, Spain, pp. 87-98, September 2009.

[23] E. Halepovic, C. Williamson, and M. Ghaderi, "Enhancing Redundant Network Traffic Elimination", *Computer Networks*, Vol. 56, No. 2, pp. 795-809, February 2012.

[24] E. Halepovic, C. Williamson, and M. Ghaderi, "DYNABYTE: A Dynamic Sampling Algorithm for Redundant Content Detection", *Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN)*, Maui, Hawaii, August 2011.