

REPOSITORY REPLICATION USING NNTP AND SMTP

Joan A. Smith, Martin Klein, and Michael L. Nelson

Old Dominion University, Department of Computer Science
Norfolk VA 23529

{jsmit, mklein, mln}@cs.odu.edu

Abstract. We present the results of a feasibility study using *shared, existing*, network-accessible infrastructure for repository replication. We investigate how dissemination of repository contents can be “piggybacked” on top of existing email and Usenet traffic. Long-term persistence of the replicated repository may be achieved thanks to current policies and procedures which ensure that mail messages and news posts are retrievable for evidentiary and other legal purposes for many years after the creation date. While the preservation issues of migration and emulation are not addressed with this approach, it does provide a simple method of refreshing content with unknown partners.

1 Introduction

We propose and evaluate two repository replication models that rely on *shared, existing infrastructure*. The premise is that if archiving can be accomplished within a widely-used, already deployed infrastructure whose operational burden is shared among many partners, the resulting system will have only an incremental cost and be tolerant of dynamic participation. With this in mind, we examine the feasibility of repository replication using Usenet news (NNTP) and email (SMTP).

There are reasons to believe that email and Usenet could function as a persistent, if diffuse, archive. NNTP provides well-understood methods for distribution and de-duping with distributed and dynamic membership. Google recently merged several historical Usenet archives to form “Google Groups,” creating a Usenet archive that covers from May 1981 to the present [1]. Although web-based bulletin-boards and blogs have supplanted Usenet in recent years, many communities still actively use (moderated) newsgroups for discussion and awareness. While not publicly archivable in the same way as Usenet, email is ubiquitous and frequent. During a 30 day test period, we averaged over 16,000 daily outbound emails through our departmental SMTP server to more than 4000 unique SMTP server destinations. Although email is point-to-point communication (unlike Usenet), given enough time, attaching repository contents to outbound emails may prove to be an effective way to disseminate contents to previously unknown locations.

These approaches do not address the more complex aspects of preservation such as format migration and emulation, but they do provide alternative methods for refreshing the repository contents to potentially unknown recipients. Although there may be quicker and more direct methods of synchronization for some repositories, the proposed methods have the advantage of working with firewall-inhibited organizations and repositories without public, machine-readable interfaces. “Piggybacking” on mature software implementations of widely deployed Internet protocols may prove to be an easy and potentially more sustainable approach for preservation.

2 Related Work

Digital preservation solutions often require sophisticated system administrator participation, dedicated archiving personnel, significant funding outlays, or some combination of these. Some approaches, for example Intermemory [2], Freenet [3], and Free Haven [4], require personal sacrifice for public good in the form of donated storage space. However, there is little incentive for users to incur such near-term costs for the long-term benefit of a larger, anonymous group. In contrast, the LOCKSS project [5] provides a collection of cooperative, deliberately slow-moving caches operated by participating libraries and publishers to provide an electronic “inter-library loan” for any participant that loses files. Because it is designed to service the publisher-library relationship, it assumes a level of at least initial out-of-band coordination between the parties involved. Its main technical disadvantage is that protocol is not resilient to changing storage infrastructures. The rsync program [6] has been used to coordinate the contents of digital library mirrors such as the arXiv eprint server but, like LOCKSS, it is based on file system semantics and cannot easily be abstracted to other storage systems. Peer-to-peer services have been studied as a basis for the creation of an archiving cooperative among digital repositories [7]. The concept is promising but their simulations indicated scalability is problematic for this model. The Usenet implementation [8] of the Eternity Service [9] is the closest to the methods we propose. However, the Eternity Service focuses on non-censorable anonymous publishing, not preservation per se.

3 The Prototype Environment

We began by creating and instrumenting a prototype system consisting of an Apache web server with the `mod_oai` module installed [10]; an NNTP news server [11], INN; two common SMTP mail servers [12], Postfix and sendmail; a news archiving tool running on a separate client machine; and a mail archiving tool also running on a separate, client system. Figure 1 illustrates the prototype environment we installed. No server was dedicated to news or mail; they also provided services to other users, including project development environments, operational software, and web services. *mod_oai* is an Apache module that provides Open Archives Protocol for Metadata Harvesting (OAI-PMH) [13] access

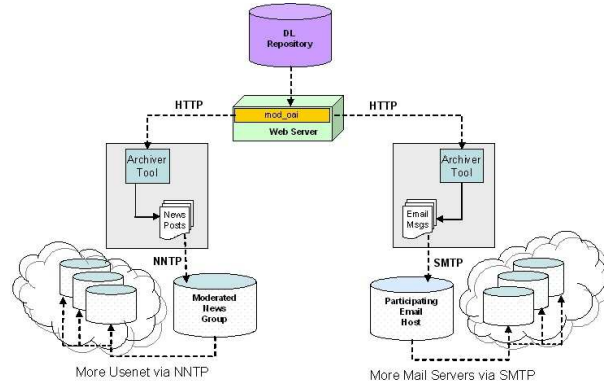


Fig. 1. The Prototype Environment

to a web server. Unlike most OAI-PMH implementations, `mod_oai` does not just provide metadata about resources, it can encode the entire web resource itself in MPEG-21 Digital Item Declaration Language [14] and export it through OAI-PMH.

We created a small repository of web resources consisting of 72 files in HTML, PDF and image (GIF, JPG, and PNG) formats. The files were organized into a few subdirectories and file sizes ranged from less than a kilobyte to 1.5 megabytes. The operating system was Fedora Core (Red Hat Linux) with Apache version 2.0.49.

For our test runs, we implemented version 2.3.5 of INN, a common, standard NNTP news server [15]. We did not change any of the default parameters: messages could be text or binary; maximum message life was 14 days; and direct news posting was allowed. Of the many SMTP mail servers available, `sendmail` and `postfix` are popular and widely deployed. We used both in our prototype system, creating the archive messages using the `postfix` (version 2.1.5) environment and sending/receiving the messages using `sendmail` (version 8.13.1).

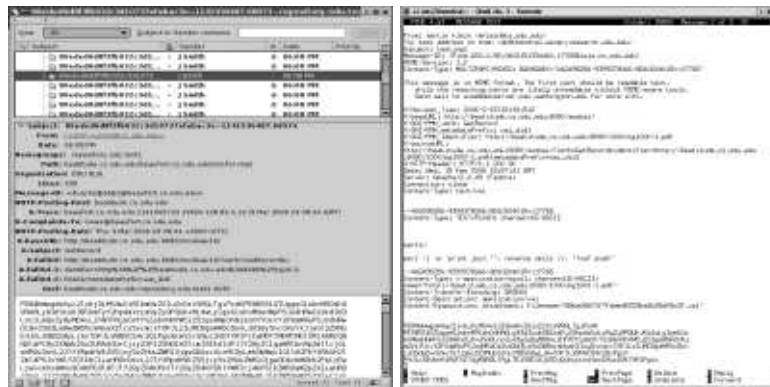
For each archiving method, we harvested the entire repository over 100 times. The archiving tools were operated from remote clients, as seen in Figure 1. Both methods used a simple, iterative process: (1)read a repository record; (2)format it for the appropriate archive target (mail or news); (3)encode record content using base64; (4)add human-readable X-headers (for improved readability and recovery); (5)transmit message (email or news post) to the appropriate server; (6)repeat steps 1 through 5 until the entire repository has been archived. Below, we discuss details of the differences in each of these steps as applied specifically to archiving via news or mail.

3.1 The News Prototype

Usenet groups exist in many formats. For our experiment, we created a *moderated* newsgroup which means that postings must be authorized by the news-

group owner. This is one way newsgroups keep spam from proliferating on the news servers. We also restricted posts to selected IP addresses and users, further reducing the “spam window.” For the experiment, we named our newsgroup “r.o.test1,” but groups can have any naming scheme that makes sense to the members. For example, a DNS-based scheme that used “repository.odu.edu” or “repository.uk.ac.soton.psy” would be a reasonable naming convention.

Using the simple 5-step method outlined above, we created a news message for each record in the repository. Figure 2(a) illustrates a sample news message and shows the X-Headers created by the tool. We also collected statistics on (a)original record size vs. posted news message size; (b)time to harvest, convert and post a message; and (c)the impact of line length limits in news posts. Our experiment confirmed the intuitive expectation that archiving via news is linear with respect to time.



(a) News Message

(b) Mail Message

Fig. 2. Sample Archival Messages

3.2 The Email Prototype

In order to get an idea of the email distribution we analyzed the outbound email traffic of the Computer Science department of ODU. Over a period of 30 days there were 505,987 outgoing emails to 4,081 unique hosts. This results in a daily mean frequency of 16,866 emails with a standard deviation of 5,147.

To illustrate how *piggybacking* of repository items is incorporated into an email message, figure 2(b) shows a screenshot of the email client Pine with the automatically attached file. Table 1 shows the actual X-Headers added to an email (or news) archival message. Notice that both human-readable OAI-PMH and web server-related headers are added to the archival messages. The goal is

```

X-Harvest_Time: 2006-2-15T18:34:51Z
X-baseURL: http://beatitude.cs.odu.edu:8080/modoai/
X-OAI-PMH_verb: GetRecord
X-OAI-PMH_metadataPrefix: oai_didl
X-OAI-PMH_Identifier: http://beatitude.cs.odu.edu:8080
/1000/pg1000-1.pdf
X-sourceURL: http://beatitude.cs.odu.edu:8080/modoai/ ?verb=GetRecord
&identifier=http://beatitude.cs.odu.edu:8080/1000/pg1000-1.pdf
&metadataPrefix=oai_didl
X-HTTP-Header: HTTP/1.1 200 OK

```

Table 1. X-Headers Added to Archival Messages

to facilitate both discovery and recovery of the archived records. How fast is the prototype and what penalties are incurred? Values obtained from our live experiments showed the expected, linear relationship between email processing and attachment size. Having created tools for harvesting the records from our sample digital library, and having used them to archive the repository, we were able to measure the results. In both cases, we found a linear relationship between the size of the repository and the time it takes to fully replicate it using either news or mail.

4 Simulating The Archiving Process

Transitioning from live, instrumented systems to simulations, there are a number of variables that must be taken into consideration in order to arrive at realistic figures (Table 2). Repositories vary greatly in size, rate of updates and additions, and number of records. With the SMTP method, we measured approximately a 1 second delay in processing attachments of sizes up to 5MB. With NNTP, we tested postings in a variety of sizes and found values ranging from 0.5 seconds (12 KB) to 26.4 seconds (4.9MB). The linear relationship between resource size and time to post the message was consistent, regardless of “slow” and “fast” periods of network speed (B). Regardless of the archiving method, a repository will have specific policies covering the number of copies archived; how often each copy is refreshed; whether intermediate updates are archived between full backups; and other institutional-specific requirements such as geographic location of archives. We call these “Sender Policies.” The receiving agent, whether it is a news or a mail server, will also have policies that impact the archiving process. Limits on individual message size, length of time messages live on the server, and whether messages are processed by batch or at time of arrival are some examples of “Receiver Policies.”

A key difference between news-based and email-based archiving is the active-vs-passive nature of the two approaches. This difference is reflected in the policies and how they impact the archiving process under each method. A “baseline” refers to making a complete replication (snapshot) of a repository. A “cyclic

Repository	R	Number of records in repository
	s	Individual record size
	$R_{\overline{s}}$	Mean size of records
	R_a	Number of records added per time unit
	R_u	Number of records updated per time unit
	$R_{\overline{g}}$	Average growth in repository size per time unit
Usenet	N_{ttl}	News posting time-to-live
	Δ_N	News posting delay (moderated groups)
	h_x	Size of X-Headers attached to news post
	t_b	Time between baseline harvests
	t_u	Time between update/addition harvests
Email	g	Granularity (applied value)
	G	Granularity (overall effect)
	T_s	Average email delivery time with attachment of size s
	T_{email}	Overall average delivery time of 1 email
	E	Attach record to every E^{th} email
General	b_{64}	Base64 encoding factor (33%)
	B	Baud rate (Network speed)

Table 2. Simulation Variables

baseline” is the process of repeating the snapshot over and over again. In between baselines, a repository could submit only updates(R_u) or new additions (R_a). Or a repository could submit an initial baseline, followed only by new or changed content. Finally, we should remember that most repositories are not static, they usually have periodic changes to existing content (R_u) as well as new additions (R_a). The changing nature of the repository must be accounted for in the archiving plan.

4.1 Archiving Using NNTP

Figure 3 illustrates the impact of policies on the news method of repository replication. A baseline, whether it is cyclic or one-time-only, should be complete before the end of the news server’s N_{ttl} , or a complete snapshot will not be achieved. Several factors affect the total time (T_{news}) needed to convert a repository to a sequence of NNTP-based news posts: the number of records in the repository (R), the number of custom-news-headers (X-headers, h_x) accompanying each news post, base64 encoding time (b_{64}), the network transfer time (B), and any posting delay (Δ_N) resulting from the moderator-approval requirement. The relationship of these factors can be seen in Equation 1.

$$T_{news} = \sum_{i=1}^R \left(\frac{(s_i \times b_{64}) + h_x}{B} + \Delta_N \right) \quad (1)$$

Encoding using base64 is fairly rapid, but it increases file size by one third, which means network speed plays a significant role in overall archiving time. NNTP is an older protocol with limits on line length and content. By converting binary

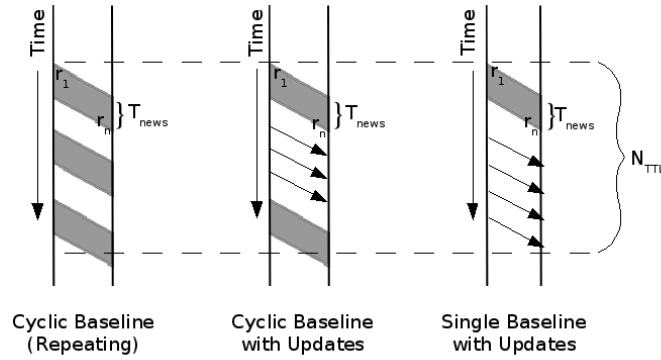


Fig. 3. NNTP Timeline for Sender & Receiver Policies

content to base64, such restrictions are overcome but at the cost of increased archive size.

4.2 Archiving Using SMTP

One major difference in using mail as the archiving target instead of news is that the mail process relies on existing traffic between the archiving site and one or more target destination sites. The two sides of SMTP-method archiving, outbound and inbound, are shown in Figure 4. As mentioned earlier, the proto-

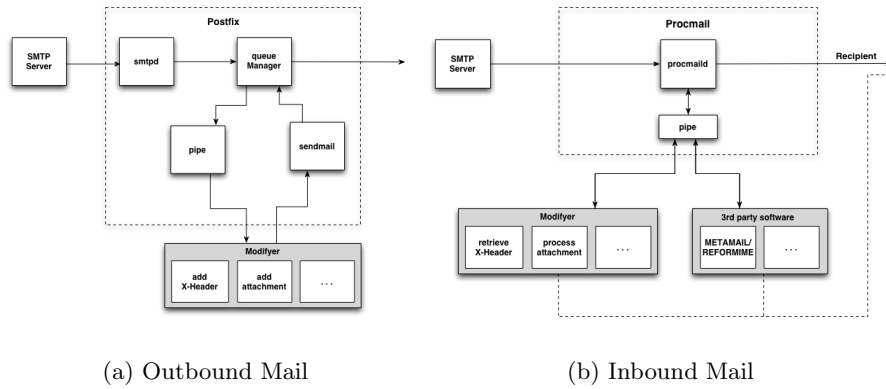


Fig. 4. Archiving Using SMTP

type is able to attach files automatically with just a small penalty. Furthermore, it is capable of distinguishing between email destinations; we are therefore able

to keep pointers to every single receiver domain and can, if desired, treat them differently. Additionally, the prototype can be fed with a granularity parameter such that only every E^{th} email is being processed by the system. Figure 5 shows the effect of increasing granularity (G) on the average time to deliver an email with a certain attachment size. Granularity appears to work as a damping factor

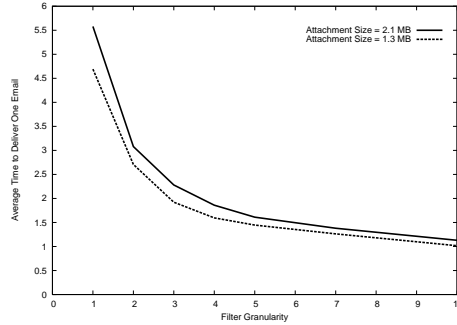


Fig. 5. Effect of Increasing Granularity on Email Delivery Time

on the average time to deliver an email. By analyzing the average delivery time of individual emails depending on the attachment size (T_s), and the granularity effect (G) from figure 5, it follows that the overall average delivery time for an email (T_{email}) with the given attachment size s and granularity g can be determined using Equation 2.

$$T_{email} = \frac{T_s * G(g)}{G(1)} \quad (2)$$

5 Results

Space limitations prevent a discussion of more than one repository profile. As such, we chose a repository profile similar to some of the largest publicly harvestable OAI-PMH repositories. For our simulations, we used a 100 gigabyte repository with 100,000 items ($R = 100000$, $R_s = 1MB$) a low-end bandwidth of 1.5 megabits per second ($B = 1.5Mbps$); an average daily update rate of 0.5% ($R_u=500$); an average daily new-content rate of 0.1% ($R_a=100$); and a news-server posting life (N_{ttl}) of 30 days. We ran the simulation for 2000 days (5.5 years).

5.1 Policy Impact on NNTP-Based Archiving

News-based archiving is constrained primarily by the receiving news server and network capacity. If the lifetime of a posting (N_{ttl}) is shorter than the archiving time of the repository (T_{news}), then a repository cannot be successfully archived

to that server. The timeline is therefore the key feasibility determinant. Figure 6 illustrates how a sufficient grace period can support different repository archiving policies. Figure 6 illustrates the simulated Repository with varying sender

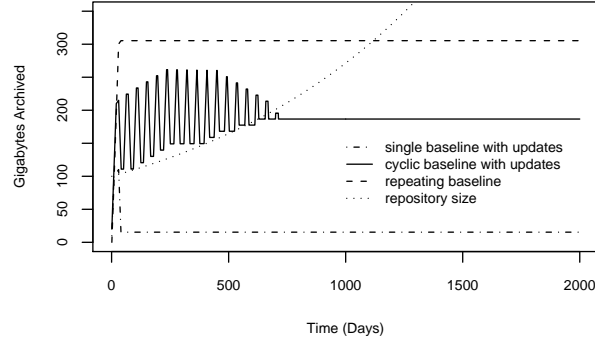


Fig. 6. Effect of Sender Policies on News-Method Archiving

policies. In one scenario, continuous baselines are transmitted (which includes both updated information and new additions with each new baseline sent). Both the “Cyclic Baseline with Updates” and the “Repeating Baseline” approaches eventually result in a steady-state amount of data existing on the news server. The second approach, “Cyclic Baseline with Updates” (Figure 6) illustrates a sender policy covering a 6-week period: The entire repository is archived twice, followed by updates only, then the cycle is repeated. This results in the news server having between one and 2 full copies of the repository, at least for the first few years. The third approach, where the policy is to make a single baseline copy and follow up with only updates and additions, results in a rapidly declining archive content over time, with only small updates existing on the server. It is obvious that as a repository grows and other factors such as news posting time remain constant, the archive eventually contains less than 100% of the library’s content, even with a policy of continuous updates. Nonetheless, a significant portion of the repository remains archived for many years if some level of negotiated baseline archiving is established.

5.2 Policy Impact on SMTP-Based Archiving

SMTP-based archiving is obviously constrained by the frequency of outbound emails. Consider the following two sender policies. The first policy maintains just one queue where items of the repository are being attached to every E^{th} email regardless of the receiver domain. In the second policy, we have more than one queue where we keep a pointer for every receiver domain and attach items to every E^{th} email going out to these particular domains. The second policy

will allow the receiving domain to converge on 100% coverage much faster, since accidental duplicates will not be sent (which does happen with the first policy). However, this efficiency comes at the expense of the sending repository tracking separate queues for each receiving domain.

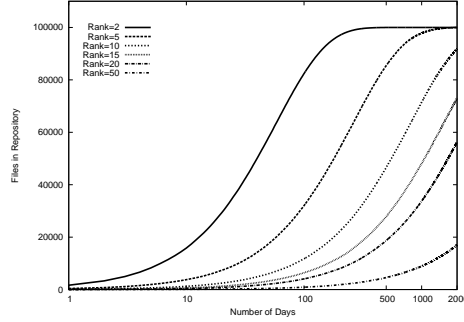
Data from the CS Department email logs showed that outbound emails follow a power law distribution. Thus, the high-ranked receiver domains achieve 100% repository coverage fairly soon and the difference between ranks two and three, for example, is not significant in either policy. However, the differences become significant as the ranking grows. For instance, Rank 20 achieved full coverage within 2000 days if the pointer was maintained, whereas without the pointer it achieved only about 60% repository replication. Figure 7(a) shows the time it takes for a domain to receive all files of a repository without the pointer to the receiver and figure 7(b) shows the same setup but with receiver pointer. In both graphs, the 1st ranked receiver domains are left out because they are the same as the local domain.

5.3 Discussion

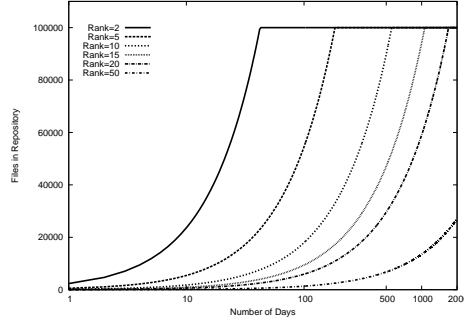
How would these approaches work with other repository scenarios? If the archive were substantially smaller (10,000 records with a total size of 15 GB), the time to upload a complete baseline would also be proportionately smaller since, as we noted earlier, the replication time is linear with respect to the repository’s size for both the news and email methods of archiving. The email approach is also dependent on the site’s email traffic volume and not just on bandwidth, since record data is only sent with every E^{th} email. A reduction in number of records therefore has a bigger impact if the repository uses the email solution, because fewer emails will be needed to transmit a complete copy of the repository. A repository consisting of a single record (e.g., an OAI-PMH “Identify” response) could be effectively used to advertise the existence of the repository regardless of the archiving approach or policies. After the repository was discovered, it could be harvested via normal means.

6 Future Work and Conclusions

Through prototypes and simulation, we have studied the feasibility of replicating repository contents using the installed NNTP and SMTP infrastructure. Our initial results are promising and suggest areas for future study. In particular, we must explore the trade-off between implementation simplicity and increased repository coverage. For SMTP approach, this could involve the receiving email domains informing the sender (via email) that they are receiving and processing attachments. This would allow the sender to adjust its policies to favor those sites. For NNTP, we would like to test varying the sending policies over time as well as dynamically altering t_b and t_u . Furthermore, we plan to revisit the structure of the objects that are transmitted, including taking advantage of the evolving research in preparing complex digital objects for preservation [16][17].



(a) Without Pointer



(b) With Pointer

Fig. 7. Domain Time To Receive 100% Repository Coverage

It is unlikely that a single, superior method for digital preservation will emerge. Several concurrent, low-cost approaches are more likely to increase the chances of preserving content into the future. We believe the “piggyback” methods we have explored here can be either be a simple approach to preservation, or a compliment to existing methods such as LOCKSS, especially for content unencumbered by restrictive intellectual property rights. Even if NNTP and SMTP are not used for resource transport, they can be effectively used for repository awareness. We have not explored what the receiving sites do with the content once it has been received. In most cases, it is presumably unpacked from its NNTP or SMTP representation and ingested into a local repository. On the other hand, sites with apparently infinite storage capacity such as Google Groups could function as long-term archives for the encoded repository contents.

7 Acknowledgements

This work was supported by NSF Grant ISS 0455997.

References

1. 20 year archive on google groups. http://www.google.com/googlegroups/archive_announce_20.html.
2. Andrew V. Goldberg and Peter N. Yianilos. Towards an archival intermemory. In *Proceedings of IEEE Advances in Digital Libraries, ADL 98*, pages 147–156, April 1998.
3. Ian Clark, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: a distributed anonymous information storage and retrieval system. In *International Workshop on Design Issues in Anonymity and Unobservability LNCS 2009*.
4. Roger Dingledine, Michael J. Freedman, and David Molnar. The free haven project: Distributed anonymous storage service. *Lecture Notes in Computer Science*, 2009:67–95, 2001.
5. Petros Maniatis, Mema Roussopoulos, T.J.Giuli, David S. H. Rosenthal, and Mary Baker. The LOCKSS peer-to-peer digital preservation system. *ACM Transactions on computer systems*, 23:2 – 50, February 2005.
6. Andrew Tridgell and Paul Mackerras. The rsync algorithm. Technical report, The Australian National University, 1996. <http://cs.anu.edu.au/techreports/1996/TR-CS-96-05.pdf>.
7. Brian F. Cooper and Hector Garcia-Molina. Peer-to-peer data trading to preserve information. *ACM Transactions on Information Systems*, 20(2):133 – 170, 2002.
8. Adam Back. The eternity service. *Phrack Magazine*, 7(51), 1997.
9. Ross J. Anderson. The eternity service. In *1st International Conference on the Theory and Applications of Cryptology (Pragocrypt '96)*, pages 242–252, 1996.
10. Michael L. Nelson, Herbert Van de Sompel, Xiaoming Liu, and Terry L. Harrison. mod_loai: An apache module for metadata harvesting. Technical report, Old Dominion University, 2005. <http://arxiv.org/abs/cs.DL/0503069>.
11. Brian Kantor and Phil Lapsley. Network news transfer protocol, February 1986. <http://www.faqs.org/rfcs/rfc977.html>.
12. Jonathan B. Postel. Simple mail transfer protocol, August 1982. <http://www.faqs.org/rfcs/rfc821.html>.
13. Carl Lagoze, Herbert Van de Sompel, Michael L. Nelson, and Simeon Warner. The Open Archives Initiative Protocol for Metadata Harvesting. <http://www.openarchives.org/OAI/openarchivesprotocol.html>.
14. Jeroen Bekaert, Patrick Hochstenbach, and Herbert Van de Sompel. Using MPEG-21 DIDL to represent complex digital objects in the Los Alamos National Laboratory digital library. *D-Lib Magazine*, 9(11), November 2003. doi:10.1045/november2003-bekaert.
15. ISC INN: Internet news. Internet Systems Consortium. <http://www.isc.org/index.pl?sw/inn/>.
16. Jeroen Bekaert, Xiaoming Liu, and Herbert Van de Sompel. Representing digital assets for long-term preservation using mpeg-21 did. In *Ensuring Long-term Preservation and Adding Value to Scientific and Technical data (PV 2005)*, 2005. <http://arxiv.org/abs/cs.DL/0509084>.

17. Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. Resource harvesting within the OAI-PMH framework. *D-Lib Magazine*, 10(12), December 2004. doi:10.1045/december2004-vandesompel.