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 USA
{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 utilize the SMTP and NNTP protocols to replicate both the metadata and the content of a digital library, using OAI-PMH to facilitate management of the archival process. 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 email 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 for smaller digital repositories that do not have the administrative resources for more sophisticated solutions.

1 Introduction

We propose and evaluate two repository replication models that rely on *shared, existing* infrastructure. Our goal is not to "hijack" other sites' storage, but to take advantage of protocols which have persisted through many generations and which are likely to be supported well into the future. 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, [1]) and email (SMTP, [2]).

There are reasons to believe that both email and Usenet could function as persistent, if diffuse, archives. NNTP provides well-understood methods for content distribution and duplicate deletion (deduping) while supporting a distributed and dynamic membership. The long-term persistence of news messages is evident in "Google Groups," a Usenet archive with posts dating from May 1981 to the present [3]. Even though blogs have supplanted Usenet in recent years, many communities still actively use moderated news groups for discussion and awareness. Although email is not usually publicly archivable, it is ubiquitous and frequent. Our departmental SMTP email server averaged over 16,000 daily

J. Gonzalo et al. (Eds.): ECDL 2006, LNCS 4172, pp. 51–62, 2006.

outbound emails to more than 4000 unique recipient servers during a 30-day test period. Unlike Usenet, email is point-to-point communication but, given enough time, attaching repository contents to outbound emails may prove to be an effective way to disseminate contents to previously unknown locations. The open source products for news ("INN") and email ("sendmail" and "postfix") are widely installed, so including a preservation function would not impose a significant additional administrative burden.

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. There may be quicker and more direct methods of synchronization for some repositories, but the proposed methods have the advantage of working with firewall-inhibited organizations and repositories without public, machine-readable interfaces. For example, many organizations have web servers which are accessible only through a VPN, yet email and news messages can freely travel between these servers and other sites without compromising the VPN. Piggybacking on mature software implementations of these other, widely deployed Internet protocols may prove to be an easy and potentially more sustainable approach to 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 [4], Freenet [5], and Free Haven [6], 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, LOCKSS [7] 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 the protocol is not resilient to changing storage infrastructures. The rsync program [8] has been used to coordinate the contents of digital library mirrors such as the arXiv eprint server but 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 [9]. The concept is promising but their simulations indicated scalability is problematic for this model. The Usenet implementation [10] of the Eternity Service [11] 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 using popular, open source products: Fedora Core (Red Hat Linux) operating system; an NNTP news server (INN version 2.3.5); two SMTP email servers, postfix version 2.1.5 and sendmail version 8.13.1; and an Apache web server (version 2.0.49) with the mod_oai module installed [12]. *mod_oai* is an Apache module that provides Open Archives Protocol for Metadata Harvesting (OAI-PMH) [13] access 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 used Perl to write our own repository replication tools, which were operated from separate client machines.

As part of our experiment, we created a small repository of web resources consisting of 72 files in HTML, PDF and image (GIF, JPEG, and PNG) formats. The files were organized into a few subdirectories with file sizes ranging from less than a kilobyte to 1.5 megabytes. For the NNTP part of the experiment, we configured the INN news server with common default parameters: messages could be text or binary; maximum message life was 14 days; and direct news posting was allowed. For email, we did not impose restrictions on the size of outgoing attachments and messages. For each archiving method, we harvested the entire repository over 100 times.

Both the NNTP and SMTP 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 email.

We took advantage of OAI-PMH and the flexibility of email and news to embed the URL of each record as an X-Header within each message. X-Headers are searchable and human-readable, so their contents give a clue to the reader about the purpose and origin of the message. Since we encoded the resource itself in base 64, this small detail can be helpful in a forensic context. If the URL still exists, then the X-Headers could be used to re-discover the original resource. Table 1 shows the actual X-Headers added to each archival message.

3.1 The News Prototype

For our experiment, we created a *moderated* newsgroup which means that postings must be authorized by the newsgroup 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 "repository.odu.test1," but groups can have any naming scheme that makes sense to the members. For example, a DNS-based

Table 1. Example of Human-Readable X-Headers Added to Archival Messages

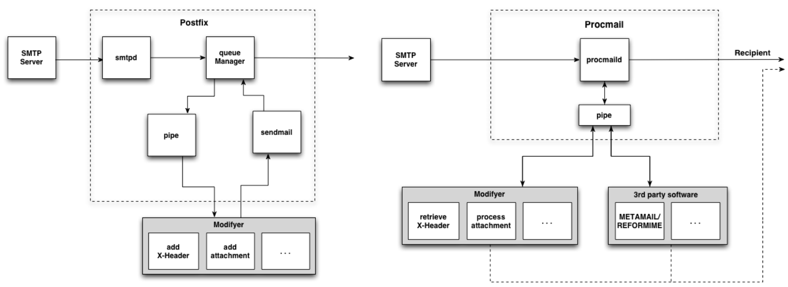
```
X-Harvest_Time: 2006-2-15T18:34:51Z
X-baseURL: http://beatitude.cs.odu.edu:8080/modoai/
X-OAI-PMH_verb: GetRecord
X-OAI-PMH_metadataPrefix: oai_didl
X-OAI-PMH_Identifier: http://beatitude.cs.odu.edu:8080/1000/pg1000-1.pdf
X-sourceURL: http://beatitude.cs.odu.edu:8080/modoai/?verb=GetRecord
&identifier=http://beatitude.cs.odu.edu:8080/1000/pg1000-1.pdf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

scheme that used "repository.edu.cornell.cs" or "repository.uk.ac.soton.psy" would be a reasonable naming convention.

Using the simple 6-step method outlined above, we created a news message with X-Headers for each record in the repository, 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 showed high reliability for archiving using NNTP. 100% of the records arrived intact on the target news server, "beatitude." In addition, 100% of the records were almost instantaneously mirrored on a subscribing news server ("beaufort"). A network outage during one of the experiments temporarily prevented communication between the two news servers, but the records were replicated as soon as connectivity was restored.

3.2 The Email Prototype

The two sides of SMTP-method archiving, outbound and inbound, are shown in Figure 1. Archiving records by piggybacking on existing email traffic requires sufficient volume to support the effort and to determine which hosts are the best recipients. Analysis of outbound email traffic from our department during a 30-day period showed 505,987 outgoing messages to 4,081 uniquehosts. A power



(a) Outbound Mail (b) Inbound Mail

Fig. 1. Archiving Using SMTP

law relationship is also evident (see Figure 2) between the domain's rank and email volume sent to that domain:

$$V_\kappa = c * (\kappa^{-1.6}) \tag{1}$$

Using the Euler Zeta function (discussed in detail in [15]), we derived the value of the constant, $c = 7378$, in Equation 1.

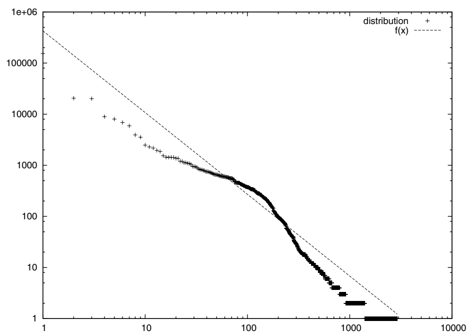


Fig. 2. Email distribution follows a power law

3.3 Prototype Results

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. How fast is each prototype and what penalties are incurred? In our email experiment, 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 processing time ranged from 0.5 seconds (12 KB) to 26.4 seconds (4.9MB). Besides the trivial linear relationship between repository size and replication time, we found that even very detailed X-Headers do not add a significant burden to the process. Not only are they small (a few bytes) relative to record size, but they are quickly generated (less than 0.001 seconds per record) and incorporated into the archival message. Both NNTP and SMTP protocols are robust, with most products (like INN or sendmail) automatically handling occasional network outages or temporary unavailability of the destination host. News and email messages are readily recovered using any of a number of "readers" (e.g., Pine for email or Thunderbird for news). Our experimental results formed the basis of a series of simulations using email and Usenet to replicate a digital library.

4 Simulating the Archiving Process

When 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. Regardless of the archiving method, a repository will have specific policies ("Sender 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 and "sleep time" (delay) between the end of one completed archive task and the start of another. The receiving agent will have its own "Receiver Policies" such as limits on individual message size, length of time messages live on the server, and whether messages are processed by batch or individually at the time of arrival.

Table 2. Simulation Variables

	R	Number of records in repository
	$R_{\overline{s}}$	Mean size of records
Repository	R_a	Number of records added per day
	R_u	Number of records updated per day
	ρ	Number of records posted per day
	N_{ttl}	News post time-to-live
	S	"Sleep" time between baseline harvests
Usenet	ρ_{news}	Records postable per day via news
	T_{news}	Time to complete baseline using news
	G	Granularity
	κ	Rank of receiving domain
Email	c	Constant derived from Euler Zeta function
	ρ_{email}	Records postable per day via email
	T_{email}	Time to complete baseline using email

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 snapshot of a repository. A "cyclic baseline" is the process of repeating the snapshot over and over again ($S = 0$), which may result in the receiver storing more than one copy of the repository. Of course, most repositories are not static. Repeating baselines will capture new additions (R_a) and updates (R_u) with each new baseline. The process could also "sleep" between baselines ($S > 0$), sending only changed content. In short, the changing nature of the repository can be accounted for when defining its replication policies.

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 finish before the end of the news server message life (N_{ttl}), or a complete snapshot will not be achieved. The time to complete a baseline using news is obviously constrained by the size of the repository and the speed of the network. NNTP is an older protocol, with limits on line length and content. Converting binary content to base64 overcomes such restrictions but at the cost of increased file size (one-third) and replication time.

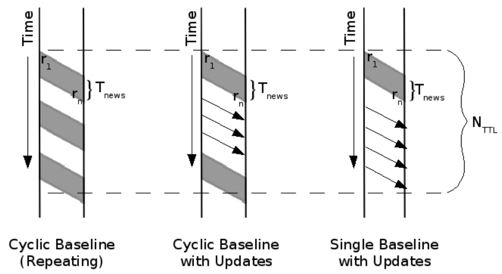


Fig. 3. NNTP Timeline for Sender & Receiver Policies

4.2 Archiving Using SMTP

One major difference in using email as the archiving target instead of news is that it is passive, not active: the email process relies on *existing* traffic between the archiving site and one or more target destination sites. The prototype is able to attach files automatically with just a small processing delay penalty. Processing options include selecting only every E^{th} email, a factor we call "granularity" [15]; randomly selecting records to process instead of a specific ordering; and/or maintaining replication lists for each destination site. Completing a baseline using email is subject to the same constraints as news - repository size, number of records, etc. - but is particularly sensitive to changes in email volume. For example, holidays are often used for administrative tasks since they are typically "slow" periods, but there is little email generated during holidays so repository replication would be slowed rather than accelerated. However, the large number of unique destination hosts means that email is well adapted to repository discovery through advertising.

5 Results

In addition to an instrumented prototype, we simulated a repository profile similar to some of the largest publicly harvestable OAI-PMH repositories. The simulation assumed a 100 gigabyte repository with 100,000 items ($R = 100000$, $R_{\overline{s}} = 1MB$); a low-end bandwidth of 1.5 megabits per second; an average daily update rate of 0.4% ($R_u = 400$); an average daily new-content rate of 0.1% ($R_a = 100$); and a news-server posting life (N_{ttl}) of 30 days. For simulating email replication, our estimates were based on the results of our email experiments: Granularity $G = 1$, 16866 emails per day, and the power-law factor applied to the ranks of receiving hosts. We ran the NNTP and SMTP simulations for the equivalent of 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. Figure 4 illustrates different repository archiving policies, where S ranges from 0 (cyclic baseline) to infinity (single baseline). The "Cyclic Baseline with Updates" in Figure 4 graphs 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

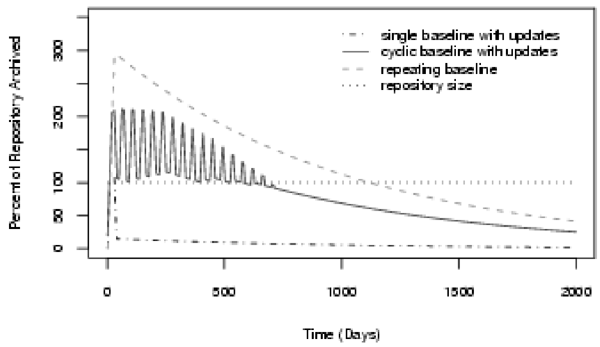


Fig. 4. Effect of Sender Policies on News-Method Archiving

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. As derived in [15], the probability of a given repository record r being currently replicated on a specific news server N on day D is:

$$P(r) = \frac{(\rho_{news} \times D) - \rho_{news} \times (D - N_{TTL})}{R + (D \times R_a)} \tag{2}$$

5.2 Policy Impact on SMTP-Based Archiving

SMTP-based replication 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.

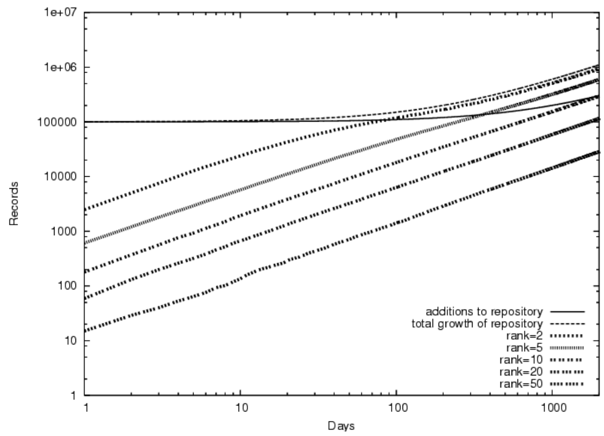
Because email volume follows a power law distribution, receiver domains ranked 2 and 3 achieve 100% repository coverage fairly soon but Rank 20 takes significantly longer (2000 days with a pointer), reaching only 60% if no pointer is maintained. Figure 5(a) shows the time it takes for a domain to receive all files of a repository without the pointer to the receiver and figure 5(b) shows the same setup but with receiver pointer. In both graphs, the 1^{st} ranked receiver domains are left out because they represent internal email traffic. Figure 5 shows how important record history is to achieving repository coverage using email. If a record history is not maintained, then the domain may receive duplicate records before a full baseline has been completed, since there is a decreasing statistical likelihood of a new record being selected from the remaining records as the process progresses. Thus, the number of records replicated per day via email ρ_{email} is a function of the receiver's rank (κ), the granularity (G), and probability based on use of a history pointer (h). That is, $\rho_{email} = c(\kappa^{-1.6}) * G * h$. If a pointer is maintained then $h = 1$; and if every outbound email to the domain is used, then $G = 1$ as well. The probability that a given record, r has been replicated via email is therefore:

$$P(r) = \frac{(\rho_{email} \times D)}{R + (D \times R_a)} \tag{3}$$

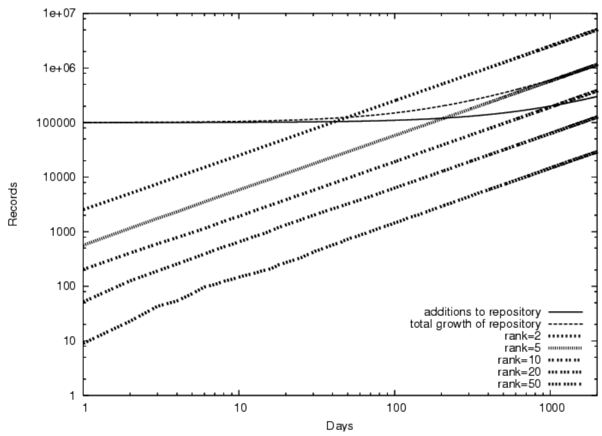
5.3 Discussion

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 replication time is linear with respect to the repository's size for both the news and email methods of archiving. The news approach actively iterates through the repository, creating its own news posts, and is therefore constrained primarily by bandwidth to the news server. Email, on the other hand, passively waits for existing email traffic and then "hitches a ride" to the destination host. The SMTP approach is dependent on the site's daily email traffic to the host, and a reduction in the number of records has a bigger impact if the repository uses the email solution because fewer emails will be needed to replicate 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. A simple "Identify" record (in OAI-PMH terms) is very small (a few kilobytes) and would successfully publish the repository's existence in almost zero time regardless of the archiving approach that was used.



(a) Without Record History



(b) With Record History

Fig. 5. Time To Receive 100% Repository Coverage by Domain Rank

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 the time between baseline harvests and transmission of update and additions. 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].

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 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.

Acknowledgements

This work was supported by NSF Grant ISS 0455997. B. Danette Allen contributed to the numerical analysis.

References

1. Brian Kantor and Phil Lapsley. Network news transfer protocol, Internet RFC-977, February 1986.
2. Jonathan B. Postel. Simple mail transfer protocol, Internet RFC-821, August 1982.
3. 20 year archive on google groups. `http://www.google.com/googlegroups/archive_announce_20.html`.
4. 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.
5. 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*.
6. 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.
7. 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.
8. 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`.
9. 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.
10. Adam Back. The eternity service. *Phrack Magazine*, 7(51), 1997.
11. Ross J. Anderson. The eternity service. In *1st International Conference on the Theory and Applications of Cryptology (Pragocrypt '96)*, pages 242–252, 1996.

12. Michael L. Nelson, Herbert Van de Sompel, Xiaoming Liu, and Terry L. Harrison. mod_oai: An apache module for metadata harvesting. Technical report, Old Dominion University, 2005. arXiv cs.DL/0503069.
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. Joan A. Smith, Martin Klein, and Michael L. Nelson. Repository replication using NNTP and SMTP. Technical report, Old Dominion University, 2006. arXiv cs.DL/0606008.
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. arXiv 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.