

Anne Wilson*
Russell K. Rew
UCAR/Unidata, Boulder, CO

1. INTRODUCTION

In the past six months Unidata's Internet Data Distribution (IDD) network has successfully delivered an aggregate of over 200 gigabytes of near real time meteorological data per day to over 130 participating institutions. The network data transmission is driven by software developed here at Unidata called the Local Data Manager (LDM). Developed over the course of the past seven years, the LDM software has proven robust, reliable and portable. Due to its success, the LDM has recently been used in several other data distribution networks.

2. CURRENT LIMITATIONS

And yet, even with significant improvements in data management algorithms, the increased volume of data available for distribution coupled with the increasing number of participating sites are pushing this software and the management of the resulting networks to their limits.

The LDM uses the notion of a "feed type" as a coarse categorization of data products. The current LDM protocol is limited to 31 feed types. With the ever-increasing amount and variety of data, this is not enough, and a protocol change to increase the limit is difficult.

The sites that participate in the IDD are configured into a tree structured topology. Paths in the topology are determined "by hand". That is, an individual assesses the quality of the connection between a new site and other possible connecting sites. If a connection is chosen, then both the upstream site and the downstream site must reconfigure their servers to establish the connection. With the growing number of participating sites and the increasing volume of data flowing along the Internet, managing this topology is becoming harder to perform.

In the LDM, downstream sites "subscribe" to products belonging to a particular feed type by using regular expressions to specify sets of products. Regular expressions are patterns used in conjunction with

pattern matching software. Regular expressions can be non-intuitive and thus difficult to read and write. Sometimes it would more convenient to simply specify unwanted products, but regular expressions do not support this well.

3. USENET AND INN

Usenet is the first news-based electronic community, and perhaps the largest. Like Unidata's IDD network, Usenet is a "logical network", a set of cooperating hosts that exchange new articles using a wide variety of communication networks. Usenet is a worldwide network, reaching every continent and nearly every country that uses computers. Well over 22,000 sites participate in Usenet. Recent examination of statistics at a large sample site shows the site receiving over 14,000,000 articles in a day for a total of over 282 gigabytes of incoming traffic alone. Usenet is a massive, unmanaged, distributed, heterogeneous network, subject to attack, and yet 90% of posted articles arrive at their destinations within an hour. Binary messages constitute the majority of traffic volume.

Network news service is based upon the Network News Transport Protocol (NNTP). Several open source implementations of this protocol exist. We are considering Internet News (INN), a popular, open source package provided by the Internet Software Consortium. This software first appeared in 1992 and has been evolving ever since.

4. SIMILARITIES BETWEEN INN AND LDM

There are many similarities between the functionality of INN and LDM. Like the LDM, INN uses a "push" approach to serving data. That is, when an article is received at a site it is automatically forwarded to all other sites who have subscribed to the group to which that article belongs. If a downstream host is unavailable, articles are spooled for later transmission. INN also provides a method to process articles as they arrive, similar to the LDM's pqact program. And, although INN has several article storage mechanisms, the fastest method is a cyclic news buffer implemented as a memory mapped file that is very reminiscent of the LDM's product queue.

5. PROBLEMS SOLVED BY INN

Use of INN addresses several limitations of the current LDM. News is organized into a virtually limitless number of hierarchically structured newsgroups which

*Corresponding author address:
Anne Wilson, UCAR/Unidata, P.O. Box 3000, Boulder,
CO 80307, e-mail: anne@UCAR.edu. The Unidata
Program Center is sponsored by the National Science
Foundation and managed by the University Corporation
for Atmospheric Research.

can be used as feed types. This not only mitigates the current limit of 31 feed types, but it also allows a finer granularity to product categorization.

News software flows through the network via a "flooding algorithm". This approach uses redundant article transmission, sending copies of an article to many sites that in turn send copies to other sites. The path taken by a copy of an article is attached to the article. A server does not propagate an article to a site that already appears in its path. The result is that an article will reach a site by the fastest route possible. This also alleviates the problem of developing and maintaining network topologies by hand. This algorithm is also robust in the face of site failure, avoiding the task of providing failover topologies.

In INN, subscriptions are defined using a syntax that allows sites to subscribe negatively to a group. Since newsgroups are hierarchically structured, it is possible to subscribe to the root of a hierarchy while omitting an entire subtree.

6. EXPERIMENTAL RESULTS

6.1 Experimental Networks

Servers were installed and configured on machines in Boulder and Washington, D.C. A new Unidata specific newsgroup, nominally called unidata.hds, was created. Graciously, Joe St. Sauver, a news administrator at the University of Oregon, agreed to propagate articles posted to this group. His hosts peer with many other news servers, providing a robust testbed.

Software for these servers include a front end program that takes a product, encodes it for transmission and adds some required header information, thereby transforming it into an "article". The output is sent to the inews program, which posts to the local news server. There is also a back end program that strips off headers and decodes the "article" back into its raw form.

This tiny network is using the existing Usenet network for product propagation. On tests involving tens of thousands of products, article latencies appear to be sufficiently small, although formal measurements must be taken.

6.2 Encoding

The NNTP protocol was originally designed for text products. Consequently, server software was also designed around transmission of text products. Yet, today, due in part to the popularity of both audio and video products, binary products constitute the majority of volume relayed via NNTP.

Use of NNTP requires encoding of binary products before transmission. Often uuencode is used for this purpose, however this can increase article size by 35%. In an effort to keep article sizes as low as possible, we

wrote our own very simple encoding algorithm. With uniformly distributed data, an increase of 5% would be expected.

Actual results on a stream of binary products showed 81% to have increased in size by 4% or less. A few products increased by up to 38%. The amount of increase slowed as the product size increased - for products greater than 10,000 bytes, the vast majority increased by only 2%.

Text streams must also be encoded because, due to their free form nature, it is not uncommon for them to contain binary characters. On a sample stream, the majority of products increased by 6% to 22%. Again, very small products had the largest increase, 16% to 24%. Most products were in the 6% to 18% range, and the few products over 10,000 bytes were in the 14% to 18% range.

7. ISSUES

7.1 The Flooding Algorithm and Security

A major benefit of using INN is the flooding algorithm. The success of the flooding algorithm depends on a large number of sites running news servers. Routing performance improves as more sites participate, as there are more choices of routes.

If we were to use INN for product distribution, we could either use the existing Usenet network, or we could develop our own separate logical network as we have done with the IDD. Using Usenet would be the best approach with respect to the flooding algorithm, because tens of thousands of sites currently participate. However, that would leave us open to attack, in the form of spamming, spoofing, and sending control messages. Malicious attack is a problem that the IDD has largely avoided up until now. In contrast, news server administrators spend much time and effort combating such attacks.

If we choose to develop our own network, with our current number of roughly 130 participating institutions, the routing possibilities are much fewer. But, 130 is only the number of institutions participating in the IDD. The actual number of institutions running LDMs is unknown, but is significantly greater than 130. If all institutions participated, the flooding algorithm might be much more successful. This would require cooperation among a variety of agencies, including NOAA, the Navy, and some commercial sites, in addition to the universities and government labs participating in the IDD.

7.2 Administrative Complexity

Many IDD sites are administered by people whose backgrounds are not in system administration. One goal of Unidata is to provide software that is portable and easy to install and maintain so that lack of system

administration experience does not keep people from participating.

INN does not fit that description. It is written in C. This means that executable code for a host must be generated from source code on a host of the same type. This requires that sites be able to compile and build their own executables, or that Unidata build those executables for them. As we support about ten different operating systems, ensuring that the code runs successfully on all platforms is nontrivial. (In contrast, Java code will run on any host.)

Furthermore, the INN package is large and complex. Being highly configurable is achieved at the price of complexity and is not for the faint of heart. The distribution comes with 24 configuration files, while the LDM uses up to four. With INN there are six different logs to monitor, while the LDM has one.

However, it is not the case that all sites that want data must run such a server. Using the NNTP protocol, sites that do not want to relay data but only wish to receive it may use a news reader package. This is a package that only retrieves articles from a server, but does not relay them. Many such packages are already available, and some are written in Java.

One solution is a two tiered approach where sites with strong administration support run the INN software and relay the data, while other sites run "reader only" software to receive products.

7.4 Article Latency

In Usenet most news articles arrive at their destination within an hour. However, such a latency would not be acceptable in the context of the IDD. In the IDD, recent statistics showed that 90% of articles arrived within ten minutes, with 73% arriving within one minute. Calculation of latencies is a key piece of research that remains to be done. An important requirement is that latencies with INN be no worse than current IDD latencies.

8. LOOKING INTO THE FUTURE

As of this writing, much research remains to be done to determine if INN is an acceptable replacement for our current LDM. However, real time data distribution is only a portion of the future of data delivery.

In the past, with lower volumes of available data and a relatively uncongested network, it was feasible and reasonable to deliver data using a push based approach. That is, data was delivered to a downstream site immediately upon reception by an upstream site.

While this approach is still applicable in many situations, in other cases the use of a pull based approach is becoming more attractive. In this approach, sites

acquire data only upon asking to receive it. Indeed, this is the basis for the Unidata THREDDS project [DOMENICO]. For example, this may be a reasonable way to distribute radar data. Due to the very large volume of radar data, pushing that entire volume to all participating IDD sites is overkill when most sites want only a very small fraction of the data.

The big open question is how to marry push based and pull based data delivery systems into a workable whole that meets the needs of the user community.

9. REFERENCES

Domenico, Ben, 2001: Thematic Real-Time Environmental Distributed Data Services (THREDDS). In *18th International Conference on Interactive Information Processing Systems for Meteorology, Oceanography and Hydrology*, Orlando, FL, January 13 - 17.