From: David M. Ihnat Date: November 18, 1988

Subject: Functional Definition: The Distributed CLOUT Machine

ABSTRACT

It has been decided that the mail service provided by /usr/group in support of the **chi.il.us** namespace will be implemented via a set of Unix machines providing a virtual, distributed gateway node to be known as *clout*. This document describes the results of the first few months of prototype operation, as well as defining the requirements, both procedural and in terms of hardware and software, for full supported implementation of this scheme.

1. INTRODUCTION

Some months ago, one of the people who were instrumental in setting up the **mn.org** moved to Chicago, and started the process of creating the **chi.il.us** domain. Shortly thereafter, AT&T announced their intention to shut off node 'ihnp4' to through traffic, creating a major disruption in E-mail connectivity in the Chicago metropolitan area, and the Midwest in general. This event prompted the Chicago chapter of /usr/group to sponsor an educational symposium on the topic of Email in general, and in particular the state of networking in the Chicago area. One of the results of this symposium was the formation of an E-mail Special Interest Group (SIG), which took as part of its mandate the setup and maintenance of the fledgling **chi.il.us** namespace.

This paper describes the current concept of a 'virtual', distributed machine—*clout*—which will act as the gateway machine for the Internet (and, eventually, for other networks such as BITNET) In particular, procedures and support software utilities that will be necessary for reliable support of the Internet-*clout* gateway are described in functional terms.

Note that it will contain much more explanatory text than is customary in a functional specification, since much of the preliminary work of meetings and discussions that usually preceeds the creation of such a document hasn't occurred among the target audience. It is expected that the final version will be considerably more succinct.

2. OVERVIEW

Conventionally, a machine running UUCP is identified to the world by its system name, which is (ideally) unique in the namespace.[1] Unfortunately, this means that any machine that is to be addressed via UUCP or domain conventions will be dedicated to that system name. This is a problem for organizations such as /usr/group, which may not(and, currently, doesn't) have the resources to own their own machine to host E-mail and networking. Private individuals and companies will, and have, volunteered to donate a certain percentage of their machine resources to help support this effort; but, understandably, they don't want to give up their own identity as a unique destination. Correspondingly, it's undesirable for /usr/group to rely upon a donated machine or resources to such a degree that the later withdrawl of such a donation can either damage reliable connectivity, or even collapse the effort, as happened with ihnp4.

2.1 MASQUERADING: CREATING A VIRTUAL MACHINE

However, there is no technical reason why a machine cannot represent itself to off-site neighbors as other than its 'real' system name. This is called *masquerading*, and is, in fact, explicitly supported as a feature of Honey-DanBer (HDB) UUCP. This allows a machine, when handling domain-related traffic, to respond and initiate transferrs as if it were the selected node. In fact, there is no actual machine *clout*. It is currently a 'virtual' machine hosted on the pre-existing machine 'homebru'; as it was previously, on machine 'jhereg'[2]

Since this scheme went into operation some months ago, it has worked servicably to support the **chi.il.us** namespace

---

[1] In practice, before domain naming, this wasn't necessarily so—using fully directed paths, there could well be two machines with identical system names, as long as they didn't share any immediate neighbors.

[2] It is, of course, possible to implement the same capability on other versions of UUCP, with varying degrees of effort. Some methods, although by no means the only ones, would be to create a special version of 'uucico', if you have source; or to actually modify the system name for the different types of traffic.

traffic. There have been occasional conflicts and problems due to the masquerading, but in all cases the existing software used to build and maintain the pathalias database has provided configuration capabilities suitable to resolve the problem.

2.2 CREATING A DISTRIBUTED VIRTUAL MACHINE

Another problem that exists is that of load distribution and cost. While one machine can certainly handle present namespace traffic, the possibility that this load may grow to unmanagable levels is a very real one--witness the growth of ihnp4 over its years of service, to the point that at the end, little or no development work could be done on the machine due to its network workload. (Despite the fact that this was the ostensible reason for its existence in the organization.)

Also, telephone rates in Illinois were recently restructured, resulting in what are effectively toll calls for any call over approximately 3 miles. This has had the effect, in many cases, of drastically raising the telecommunication costs associated with the UUCP network; often, to the point that donors of machine resources have to re-evaluate their commitment to do so.

2.3 MULTIPLE CLOUTS

2.3.1 UUCP

The only thing uniquely linking a machine system name—such as CLOUT--to a physical site is the telephone number. If two machines, sitting side-by-side, have different phone numbers in their L.sys or Systems file for the same destination, then a piece of E-mail destined for that destination would be delivered to different physical machines by each of these systems. Provided that this is appropriately administered, this allows the establishment of any number of virtual *clout* nodes, physically distributed about the Chicago metropolitan area. Thus, when machines are accepted into the **chi.il.us** namespace, they may be 'assigned' to the distributed node that is closest to them physically. This has the dual benefit of allowing the *clout* administrators to minimize telephone costs, and balancing the workload on any particular node of *clout*.

2.3.2 The Internet

As far as the gateway with the internet, for every instance of *clout*, there must be an Internet host machine. For the internet, the mechanism used to describe connectivity is the MX record; an example follows.

```
Clout.CHI.IL.US    IN   MX      10   oddjob.uchicago.edu
                   IN   HINFO   ATT-7300 UNIX

Chinet.CHI.IL.US   IN   MX      10   oddjob.uchicago.edu
                   IN   HINFO   ATT-3B2 UNIX

Homebru.CHI.IL.US  IN   MX      10   oddjob.uchicago.edu
                   IN   HINFO   ATT-7300 UNIX
```

Notice that, simply enough, an entry consists of an associated cost, and the Internet name of the server for that entry. In the above examples, then, any traffic destined for any one of the entries will be directed to **oddjob.uchicago.edu**. Fortunately, the capability exists to describe multiple servers for a given destination; for instance, **chinet.chi.il.us** could also have an MX record naming **foobar.rutgers.edu** as a machine which knows how to resolve the address (provided, of course, that this had been set up on machine 'foobar'...) The last link in the series of features that make this entire scheme feasible is the capability to, by judiciously setting the costs associated with multiple Internet servers, allow the automatic selection of appropriate servers in order, for instance when the primary machine is unreachable for some reason.

3. THE PROBLEMS

This is the purpose of this paper. It has been shown that, separately, the mechanisms exist to allow the creation of a virtual, distributed node, and that there are valid benefits to be derived from this. However, there are problems introduced that must be addressed:

- *Node Administration*
  Each node of the distributed machine requires some local administration. Most particularly, this consists of maintaining a valid pathalias database; maintaining a valid L.sys or Systems file section of *clout* intra-machine connections; and monitoring the mail throughput to catch any problems or constipation as soon as reasonably

possible. Of these, the first two may be automated, in conjunction with the master node; while the latter is a procedural requirement that the system administrator be alert and active in system monitor functions.

Note that an L.sys or Systems file record set should be generated for each participating Internet site, and appropriately ordered from most preferable connection to least preferable connection. This will allow each *clout* node to recover from temporary inability to contact its primary Internet server; this ordering and record set generation should be provided by remote request by the master *clout* node.

- *Internet MX Record Administration*
  As sites are added, or (more rarely) reassigned or deleted, information necessary to permit inclusion of MX records must be mailed to the appropriate recipient. Each *clout* node administrator must have the capability of determining the existence of any set of MX records without manual intervention of the site administrator of his/her associated Internet host, as well as receiving a current listing of such records. This may be accomplished by actions ranging from the minimum effort of providing a login to approved node administrators on the Internet site (if acceptable to the governing organization of that site), to providing a secure, remotely-executed program which may extract and return such information via a uuxqt request from authorized user/sites. It may well be, for the initial implementation, that the individual login is the simplest solution, since the number of Internet sites participating is

  small, as is the number of *clout* nodes. However, the requirements should be set forth for the secure server program in anticipation that granting of a login on all involved Internet sites may not be feasible.

  Also note that there should be an MX record set

generated for every participating *clout* node, such that the cost associated causes a prioritization scheme allowing the Internet hosts to recover from a temporary inability to contact its primary *clout* node.

- *Connectivity Configuration*
  This is more than just keeping a list of sites. Ideally, each *clout* node should be affiliated with the Internet server that is physically closest to it, to minimize telephone costs. In addition, ancillary information that is not normally part of either the UUCP Systems or L.sys files or the MX database is necessary. This minimally includes:
  - Contact persons at affiliated Internet hosts and *clout* nodes
  - Generated L.sys and MX record sets,
  - Generation of pathalias databases for each *clout* node which needs them
  - Any information which may be deemed necessary for *clout* distributed administration subsequent to the initial implementation.
- *Master Node Administration*
  For a number of reasons, it is desirable to specify one node as the "master" *clout* node. It should have somewhat more disk space and CPU cycles available than a 'normal' *clout* node, for administrative use. The entire set of pathalias maps should be maintained online, as well as the superset of all the information mentioned in the Connectivity Configuration section of this document. All associated utility scripts, programs, and data must not only reside on the master node, but also appropriate remote execution permissions should be available to the remote *clout* node administrators to allow automated inclusion of information updates, requests for new UUCP information for their dependent sites, and other information deemed appropriate for dissemination in such a manner.

Importantly, when given the pathalias map entry for a new site, support software (to be developed) should be able to determine priorities and assignments such that L.sys/Systems file record sets and MX record sets may be generated linking the new site to its appropriate *clout* node.

Since the entire pathalias source map base is rather large, while a generated pathalias database is only about 500-600K, each *clout* node must maintain a database, but if space is tight, they should be able to remotely request generation and delivery of a current database for their node from the master node. Even when not requested, this should be automatically generated at some reasonable interval and sent to sites which require this service.

For each of the issues mentioned above, a detailed section describing the procedural and/or software solutions and approaches to be followed and/or used by *clout* master and dependent node administrators will be provided. Taken together, this will serve as the basis for a functional document detailing final implementation of the distributed virtual clout machine.

--dmi David M. Ihnat