

1981

The CSNET User Environment

Peter J. Denning

Douglas E. Comer

Purdue University, comer@cs.purdue.edu

Report Number:

83-456

Denning, Peter J. and Comer, Douglas E., "The CSNET User Environment" (1981). *Department of Computer Science Technical Reports*. Paper 375.
<https://docs.lib.purdue.edu/cstech/375>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

THE CSNET USER ENVIRONMENT

Peter J. Denning
Douglas E. Comer

Department of Computer Science
Purdue University
CSD-TR-456

July 8, 1981
(revised July 27, 1981)

INTRODUCTION

CSNET is a computer network linking computer science (CS) research groups in the United States. It was funded by the NSF in 1981 as a five-year project; most of its software will be available by 1984 and most of its management and costs will be assumed by a consortium of the CS research groups by 1984. By providing sophisticated message and file transport facilities, CSNET aims at a quantum jump in information dissemination and productivity in the CS research community.

CSNET is actually a logical network encompassing three existing communication networks (Arpanet, Telenet, and telephone lines). CSNET does not own communications facilities. *CSNET provides a common set of conventions for naming senders and recipients and for transferring information between them.* CSNET also provides services not present in these other networks, including interfaces between Berkeley UNIX and the protocols used in the Arpanet and Telenet, a high-level user environment, a Nameserver for locating users, and software distribution services.

This document describes the architecture of the CSNET environment and of the Nameserver, the two most important aspects from the viewpoint of the user community.

DEFINITIONS

The Arpanet is the computer network owned and operated by the Defense Advanced Projects Research Agency (DARPA); it links over 100 sites, including 15 universities, whose research mission is of interest to the Department of Defense (DoD) or the Department of Energy (DoE). The actual network links machines called Interface Message Processors (IMPs) or Terminal Interface Processors (TIPs) by high bandwidth leased telephone lines (typically 56Kb); computers at Arpanet sites are connected to IMPs or TIPs.

Telenet is a public packet data network that allows registered users remote login at 300 baud or 1200 baud to distant sites. Telenet permits the user to open a "virtual circuit" (communications path) to the remote site, then to send and receive information as if he were logged in at a terminal on that computer. The user's account on the remote computer is charged for connect time and packets. Hardware now coming to market permits VAX/UNIX systems to connect directly to Telenet via the X.25 protocol and exchange information at much higher rates host-to-host (e.g., 9600 baud).

Phonenet is provided by CSNET for those sites not connected either to Arpanet or Telenet. Relay computers presently at the University of Delaware (Newark, Delaware) at the Rand Corporation (Santa Monica, California) dial up client computers at regular intervals to log in and exchange mail and messages (at 300 or 1200 baud). The Phonenet relay machines are connected to Arpanet and Telenet. The software that exchanges and relays messages is called the Multichannel Memorandum Distribution Facility (MMDF). The users must pay for long distance telephone charges.

A *CSNET site* is an institution housing one or more machines that are connected to CSNET. The machines of a site may be interconnected by a local network. A *CSNET host* is a machine or subsystem of a machine connected to Arpanet, Telenet, or Phonenet. The same machine may comprise more than one host. Therefore, a site is a set of one or more hosts at the same institution. CSNET sends mail to sites; sites distribute mail among local hosts.

A *CSNET user* is an individual (or group of individuals) registered in the CSNET registry; a user can directly send and receive messages via CSNET. Associated with each use is a unique *inbox*, which is a receptacle for messages sent to him. CSNET guarantees the integrity of messages until the user successfully moves them from his inbox to his local store.

The rest of this document outlines a set of requirements and functional specifications on the user environment. It is interesting that some of the requirements are similar to those imposed by the principles of data abstraction. We can therefore specify the user environment as an abstract machine.

REQUIREMENTS

1. *Coherence and Adaptability:* CSNET should integrate many disparate low-level mail systems into a uniform user environment. It must work well over a large range of physical networks, operating systems, and user communities. It should not require complicated adjustments when component systems change size or shape or when load patterns and connections on component networks change.
2. *Configuration Independence:* The CSNET abstract machine should look the same to users, no matter what host installs it. The software should be highly modular to allow for unforeseen changes and to permit new physical networks and operating systems to be integrated easily.
3. *Integrity and Sharp Boundaries:* CSNET should have well defined boundaries. Integrity of CSNET service should not depend on the correctness of its clients--a client's error hurts at most himself. An inbox should be at every point where messages enter or exit CSNET. CSNET should guarantee that mail gets to the intended recipient and that undeliverable mail is returned to the sender with appropriate error notices.
4. *Address Independence:* A single notice to the CSNET address registry should be sufficient to change a user's physical address. There should be no requirement to notify multiple sites or update any local mailers. This implies that each user must be assigned a unique identifier, called a *UName*, on registering with CSNET, that each site is assigned a unique identifier, called an *SName*, and that the CSNET user environment must map UNames to SNames. UNames can be symbolic strings meaningful to the users; the full legal name of the registrant would be a good choice (e.g., Comer's UName would be "Douglas Earl Comer"). Similarly, SNames can be symbolic (e.g., Comer's inbox will be at SName = "purdue").
5. *Local Abbreviations:* Users will employ local abbreviations, called *aliases*, in place of UNames for sending mail. This means that the local CSNET environment must be able to map abbreviations to UNames and to deal with mailing lists denoted by local abbreviations. Although local sending abbreviations are chosen separately by each user, receiving abbreviations (if allowed) must be site-controlled.

6. **Transport Generality:** CSNET makes no assumptions about message content. (Messages bodies are totally uninterpreted.) Users can mail any file, whether it contains text, data, cipher, or arbitrary bits. The software will deliver an exact copy of the file at the destination. The transporter will map unencrypted text files to the character set of the destination when instructed to do so.
7. **Single Inboxes:** Each registered user has one inbox. Everything sent to him from within the local network or from CSNET arrives in that inbox. A user's inbox is at any given time assigned to exactly one site known to CSNET; the site assigns the inbox to one of its hosts. A user desiring several inboxes must register once with CSNET for each inbox.
8. **Local Delivery:** CSNET sends incoming mail to a distribution point, called a *Mailserver*, at each host. The Mailserver sorts the mail into local user mailboxes. Delivery "c/o postmaster" at a site should be included to permit human intervention.
9. **Directory Assistance:** CSNET maintains a *Nameserver* (NS) to provide directory assistance to users and addresses to CSNET mailer processes. Each record in the NS database contains the full name and address of a user, a UName, the current SName of his box, and a list of keywords supplied by the user giving additional information about him. (Each record is uniquely identified by the UName.) Notices of address changes for registered users are entered in the NS by the CSNET administration; users can change their own keywords at any time. A given user may be registered more than once.
10. **Types of Mail:** The CSNET environment should support various types of mail, e.g.,
 - a. Regular mail: the usual form, no restrictions on content or distribution.
 - b. Registered mail: the sender gets an acknowledgment as soon as the mail is delivered to the destination inbox.
 - c. Encrypted mail: mail can be encrypted using either a public key system or the Data Encryption Standard (DES).
 - d. Signed mail: mail can be digitally signed.
 - e. Personal/Confidential mail: the receiver cannot send copies of personal mail to anyone; he can send copies of confidential mail only to an immediate group. [May be difficult to implement.]
 - f. Mail with attachments: mail can contain attached files; the recipient can put these in his local store without having to read them.
11. **Mail Facilities:** These are the operations implemented by the local mail environment, such as copying, blind copying, stating the subject, automatically keeping copies of sent mail, ability to see mail headers, editor invocation, and so forth. CSNET must choose a mail preparation and reading environment.
12. **Authentication:** CSNET attempts to authenticate senders (before accepting their messages) and receivers (before giving access to inboxes). [Because we have little control over the security mechanisms of sites, this function may be better served with encrypted mail.]

1. The File Transfer Protocol (FTP) of the Arpanet requires one communicant to read or write files in the distant system, operations that must conform to the distant system's file access controls. FTP violates the sharp boundary principle of CSNET.

A MODEL FOR ADDRESSING

The first two requirements (on addressing) imply a model of the CSNET environment. Because CSNET is a distributed system with many components, it is important to minimize the amount of changeable information that is handed out and stored in local tables. Unique identifiers are invariant under changes in the owner's physical address.

Figure 1 shows the (logical) form of a message: header, body, and attachments. The header contains the (authenticated) sender UName, a recipient list (one or more UNames), a postmark identifying date, time, and host of the transmission, and pointers to the body and attachments. The body is usually text to be read by the recipient. The attachments are arbitrary files.

Figure 2 illustrates a high-level view of an entry in the Nameserver data base. It contains the UName, the associated SName, the authentication password (if any), a timestamp, and a set of keywords. Entries are distinguishable solely by UNames. The timestamp denotes the last time the record was updated; it can be used to tell whether information encached at other sites is still current. The keywords are supplied by the user; they are used by a database system to help locate user(s) of given characteristics. The user's address and telephone number should be among the keywords.

Figures 3 and 4 illustrate the facilities of the Mailserver at a CSNET host. The user's Mail Environment is not part of the Mailserver. Figure 3 shows the levels of addressing through which a user's message must pass en route from the Mail Environment to the Physical Network (Arpanet, Telenet, or Phonetnet). The Mail Environment sends the message file (F) with the local name (N) to the Local ID Map. The Local ID Map maps the name to a UName (X) using an alias table. The Address Map maps the UName to the SName (A) of the addressee. The Netmail transmitter maps the SName to the physical address (P) on a network containing the target site and sends the message over that physical network.

Figure 3 also shows (with dashed lines) command paths used by the various levels. The *set alias* command is used to associate an abbreviation with a CSNET identifier. To obtain a UName, the Local ID Map may interact with the Nameserver via the *getuid* command. The Address Map maintains a cache of recently used UName-SName pairs; if the cache contains no correct address for a given UName, the Map interacts with the Nameserver via the *getaddr* command. The entries of this cache can actually be stored in the alias tables. The Netmail Transmitter uses a cache of common CSNET-Physical address pairs to avoid unnecessary interactions with the Nameserver.

All mail sent on by CSNET Mailservers contains the UNames of both the sender and receivers. This permits undeliverable mail to be returned; it also permits the Mailserver to enforce the requirement that mail marked "personal" by the sender cannot be remailed.

Figure 4 augments Figure 3 with additional function for receiving mail. Mail arriving from CSNET is delivered to the Netmail Receiver, which uses a Received Address Map to determine a local address corresponding to the recipient UNames attached to the mail. There is one entry in the Received Address Map for each CSNET user at that site. Each entry is of the form (UName, list), where the list contains one or more local user identifiers or an indicator that a given user is no longer at the site. The Local Network Mailer delivers the mail across a local network to the user. The Local Network Mailer may deliver the mail to a relay process for transmission back out across CSNET--e.g., in case the mail is undeliverable. The

Local ID Map is refined to include a Local Mail Switch that gates mail from a user to the local network if that mail is for some local addressee.

Figure 5 is a diagram of the main components of an addressing scheme satisfying the above requirements. Each user at each site has an *alias table* whose entries are of the form (N,L,X,A), where

- N is a string giving a local abbreviation for some user;
- L is a bit indicating whether the addressee is at a different site or at the same site (but possibly a different host);
- X is the UName of the addressee; and
- A is the most recent SName used to reach X.

The entry A is present only as a performance optimization; it acts as a cache to reduce the number of messages sent to the Nameserver in search of addresses. (In other words, the address cache used by the Address Map of Figure 3 is the set of A-fields in alias tables.) The L bit tells whether an address is local or part of CSNET. Each site can determine how much storage to allocate for alias tables and caches.

The alias table can be extended to store timestamps indicating when each CSNET SName was encached and keywords that were used to locate X in the Nameserver database.

The Nameserver maintains a central map that gives, for each UName X, an SName A. (The map is constructed from the registry.) The SName is the identifier of a site containing the inbox of X; this inbox can be a gateway into a local network or another physical network. A registered individual has exactly one CSNET SName even if his site is accessible through several physical networks.

The SName A is used by the Nameserver to locate an address list containing one or more physical network addresses for site A. The entries in this list are of the form (NET,P), where

- add space*
- NET is the name of a physical net that reaches A; and
 - P is an address on that physical net.

A physical address is a character or bit string recognizable by the corresponding physical network. (Arpa internet address are examples.) The physical network handles the routing of the message to its destination. If individuals at a given site are partitioned into access classes, a separate SName can be assigned to each class; this permits, for example, a site to distinguish users with access to Arpanet from users with access to Telenet.

OPERATIONS

We envision operation as follows. To make an entry in his alias table, a user issues the command

set alias N keywords,

This command sends its site SName (S) and the keywords to the Nameserver. The Nameserver queries its registry for a user uniquely identified by the keywords. On finding an entry, it obtains the UName (X) and the SName (A) for that user's inbox. Finally, it returns (X,A) to the requesting site, where the alias command stores the entry (N,1,X,A) in the requester's alias table.

To send a file F to the addressee whose local name is N, a user (with UName U) issues the command

send F to N-list.

This command sends (N-list, U, F) to the Local ID Map. The Local ID Map looks up the SName A in the user's alias table for each N of N-list, thereby constructing a corresponding A-list, and sends (A-list, U, F) to the Netmail Transmitter. For each entry in A-list, the Netmail Transmitter looks up an entry (A,P) in its recent address cache (or obtains P from the Nameserver) and sends (P, U, F) out on the physical network for which P is an address. If later, the message (P, U, F) is returned by the target site marked "addressee unknown", the Netmail Receiver can pass it to the Local ID Map; the Local ID Map will attempt to obtain a correct address from the Nameserver, update the alias table of U, and retransmit.

The alias table satisfies Requirement 4, that no user be forced to remember UNames. The user can employ any convenient local abbreviation for any individual to whom he wishes to send mail. The scheme also permits two or more users (at the same or different sites) to assign different local strings for the same addressee.

A change in a user's physical address change the SName (A) in the Nameserver's Map. Such changes will invalidate the A entries in alias tables that lead to him. These invalid entries will be detected the next time that address is used. The entries in A fields could be automatically discarded if not used for more than a certain interval; this interval is chosen to trade off between the overhead of sending mail to erroneous addresses and the overhead of asking the Nameserver for the current address. The overheads of registering new users and informing the name server of address changes will not be significant in CSNET.

In principle, any invariant string can serve as a unique identifier. The full legal name of the registered user would be an excellent choice. But a string like

alias@host

would be unsuitable because it is not invariant under changes in the local user identifier, the site identifier, or changes in address. Similarly a string like

UName@SName

is unsuitable because it fixes the binding of SName to UName too early and forces the users to work with inconveniently long strings. Public encryption keys are also unsuitable because they can be changed at any time.

Some mailing schemes (e.g., uucp) permit users to send mail by specifying their own routing paths. Such schemes are incompatible with some of the physical networks (e.g., Arpanet), which contain dynamic routing algorithms.

EXAMPLES

Figure 6 illustrates that CSNET can be fully transparent to any of the users of an existing network. A user already on the Arpanet, for example, can set aliases identical to the Arpanet symbols to which he has become accustomed. The translation from alias to physical address will be an identity map in this case. The gain is, of course, that the Arpanet user obtains all advantages and services of CSNET without changing the appearance of his local environment.

Figure 7 shows how CSNET will handle sites containing multiple hosts. (If two hosts are on a local network, CSNET would not be the most efficient means of exchanging mail between them.) If mail arrives at Host 1 from Arpanet intended for user Y whose inbox is Address Map directing it to forward the mail over the local network to the Mailserver at Host 2. The Mailserver at Host 2 will deposit the mail in Y's inbox. This permits the site to make local adjustments (e.g., moving Y's inbox to Host 1) without having to notify the Nameserver.

The proposed architecture handles temporary address changes smoothly. Recently Bill Kern moved his inbox from Rand-ai to Utah-20 because Rand-ai was down; but the change of address notices were not received by some users before they sent him mail at Rand-ai. Some of the mail sent to Rand-ai was lost. Shortly thereafter, Kern put his inbox back at Rand-ai--and some mail sent to Utah-20 was lost. With the proposed architecture, Kern would simply notify the Nameserver database; CSNET would guarantee that his mail reached his new inbox. There would be no need to notify potential senders of the change, nor would any confusion or lost mail ensue.

The proposed architecture also handles delivery to people at sites who are not registered users. The mechanism is to associate a special inbox with the site. A "Postmaster Process" can scan this mail--e.g., by reading extra header information embedded in the body--and attempt to deliver it to unregistered local users. The Postmaster Process can call for human intervention if necessary. If the mail is undeliverable, it can be returned to the sender using the ordinary return mail functions.

This also works for internet gateways. Each gateway looks like an inbox to CSNET. Mail delivered to it by CSNET is received by a process operating in the other net's environment and sent to its final destination. Mail delivered to it by the other network is processed by a Netmail Receiver process and forwarded on to its final CSNET destination. It is the responsibility of each network to implant "return-to" information in a message so that undeliverable mail in the other net can be returned to sender once reentering the home net. This clean interaction is made possible by the Sharp Boundary principle of CSNET.

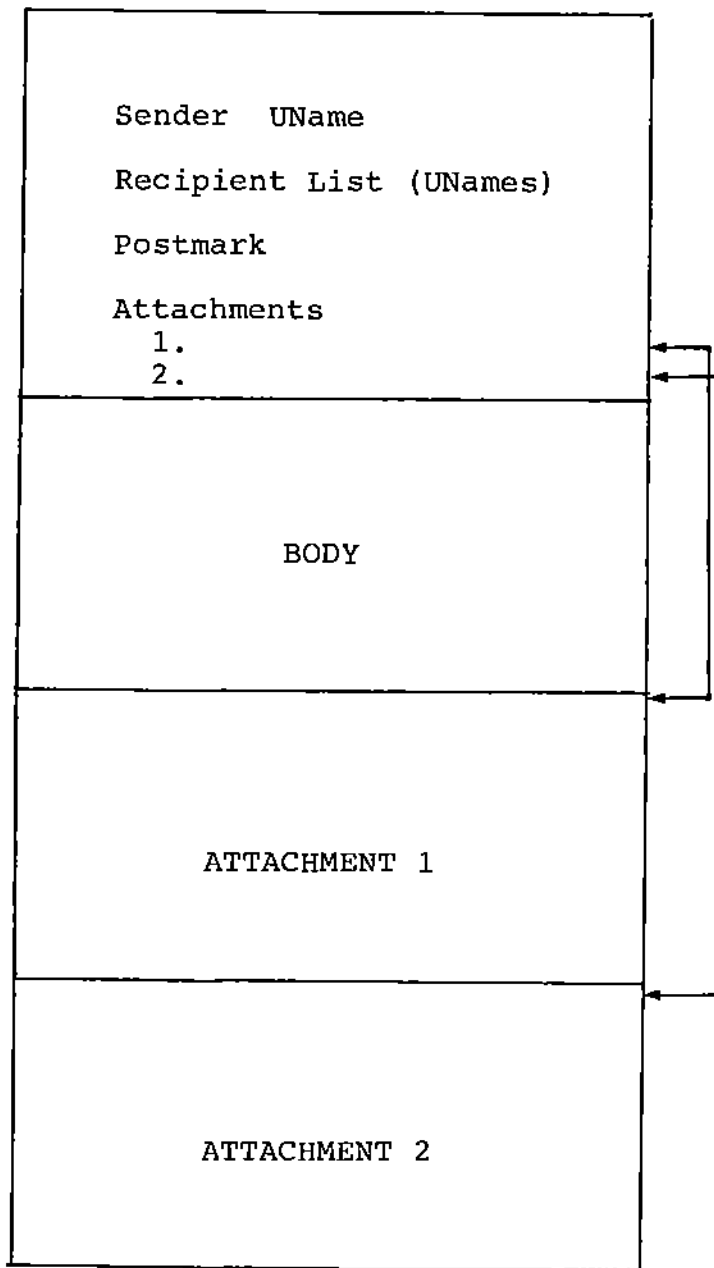


Figure 1: Logical Format of CSNET message.

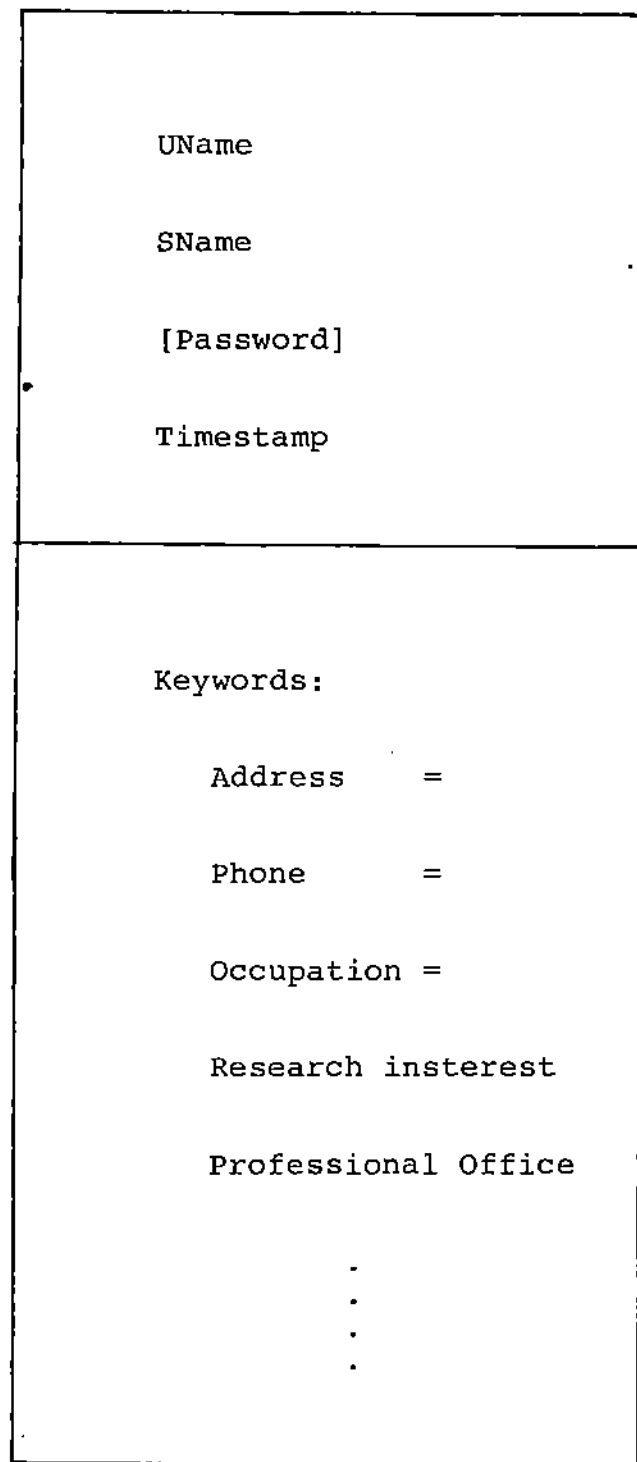


Figure 2: Entry in Nameserver registry.

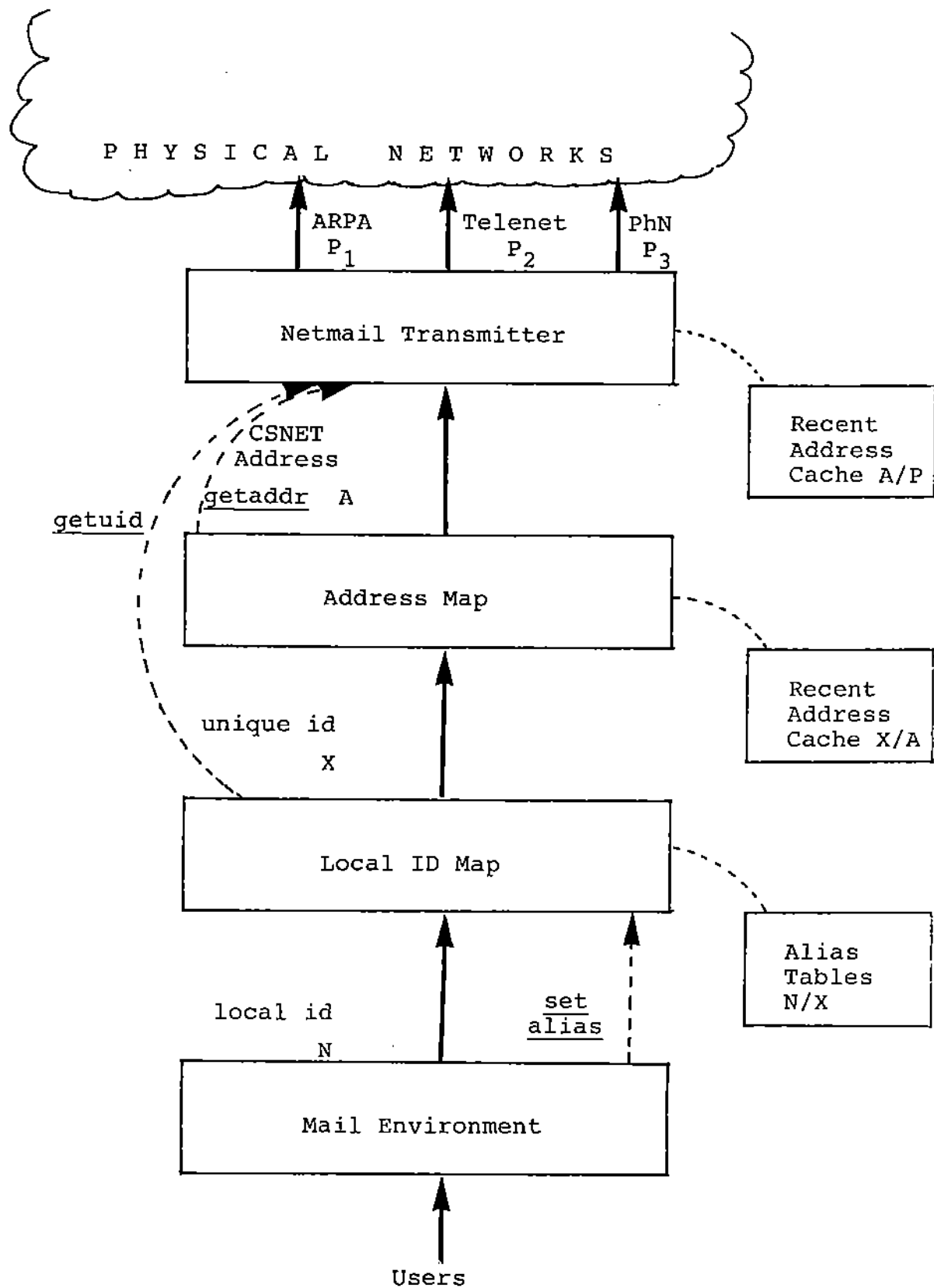


Figure 3: Functional components of CSNET environment (transmitting).

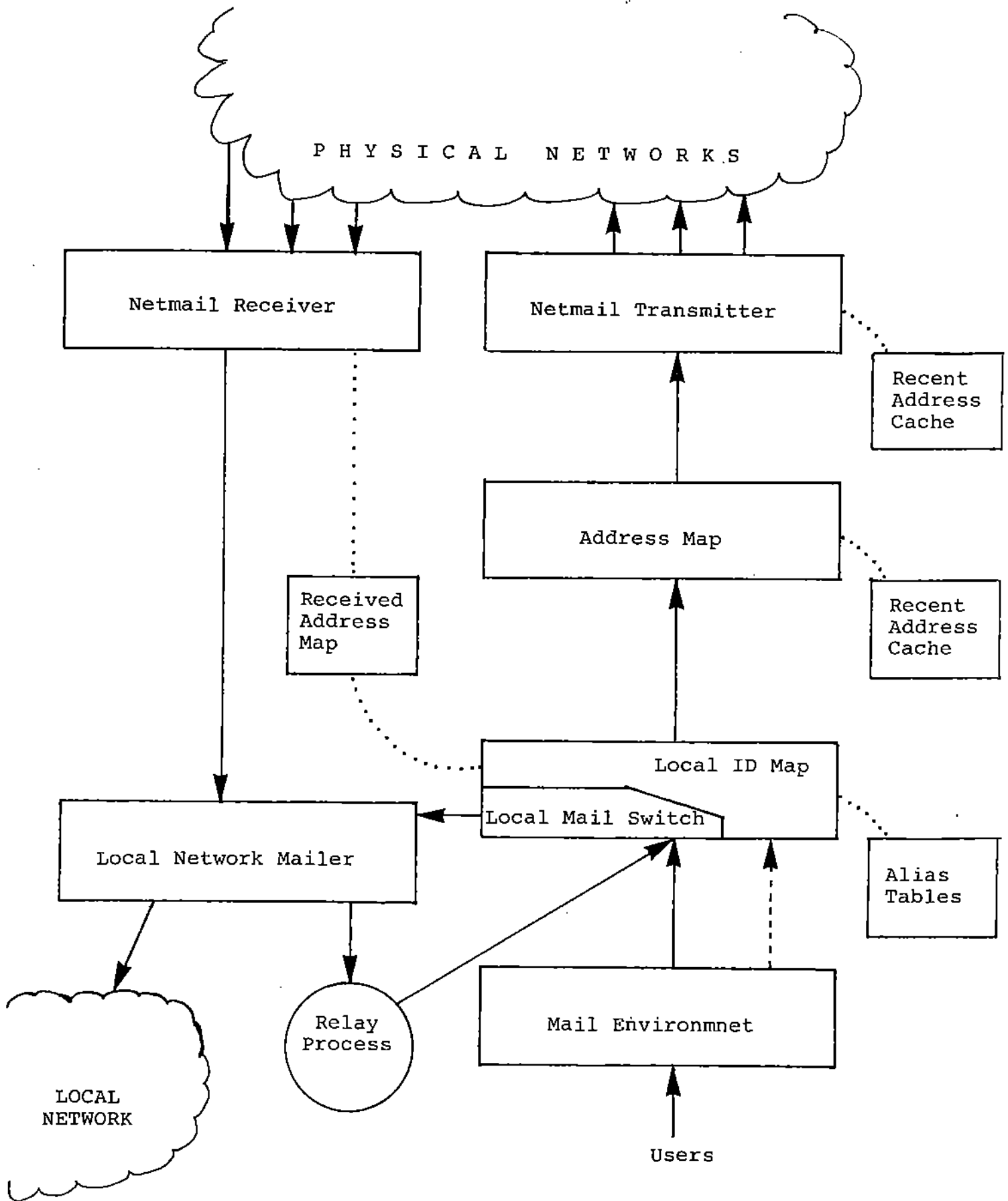


Figure 4: Functional Components of CSNET environment (receiving).

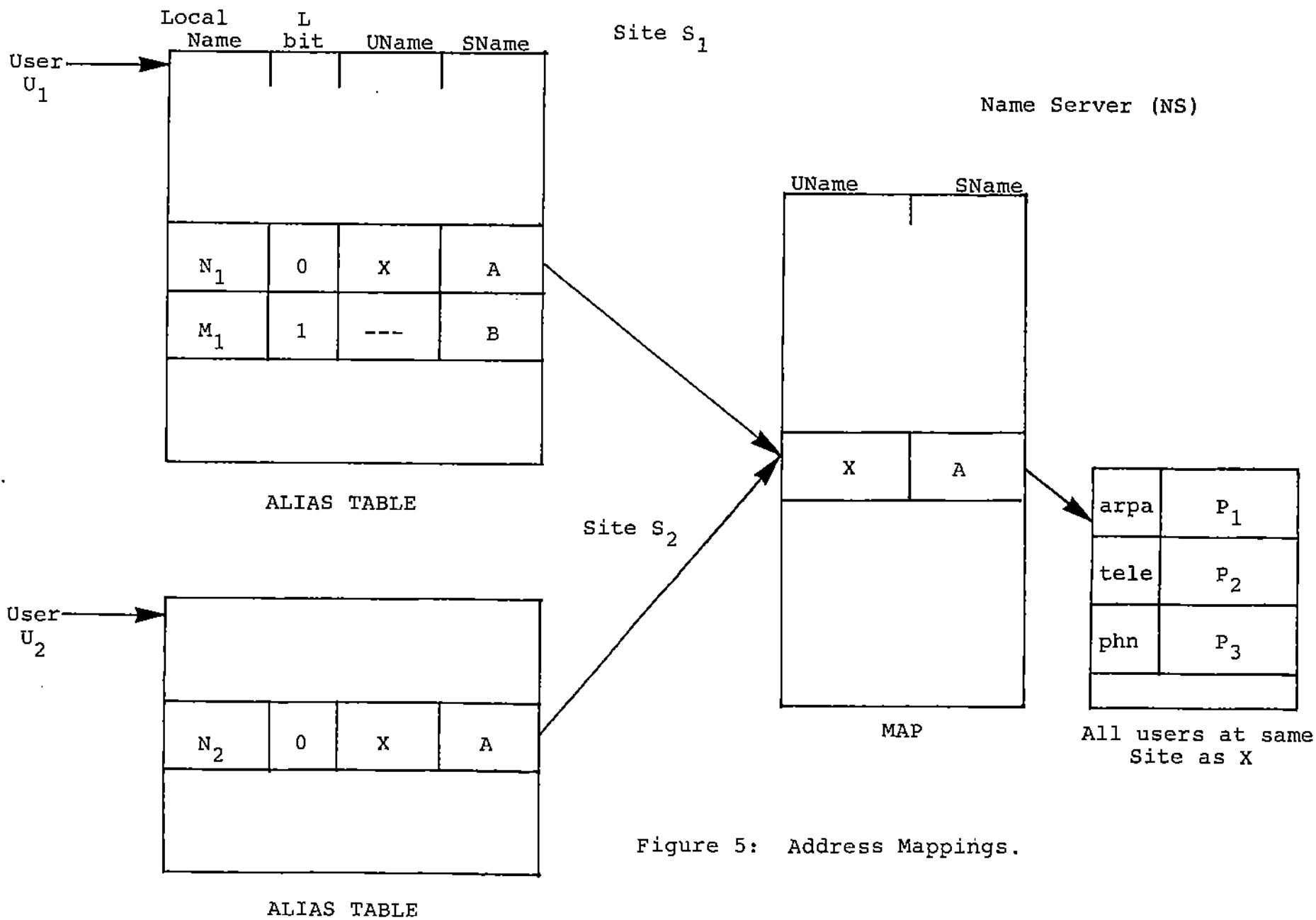


Figure 5: Address Mappings.

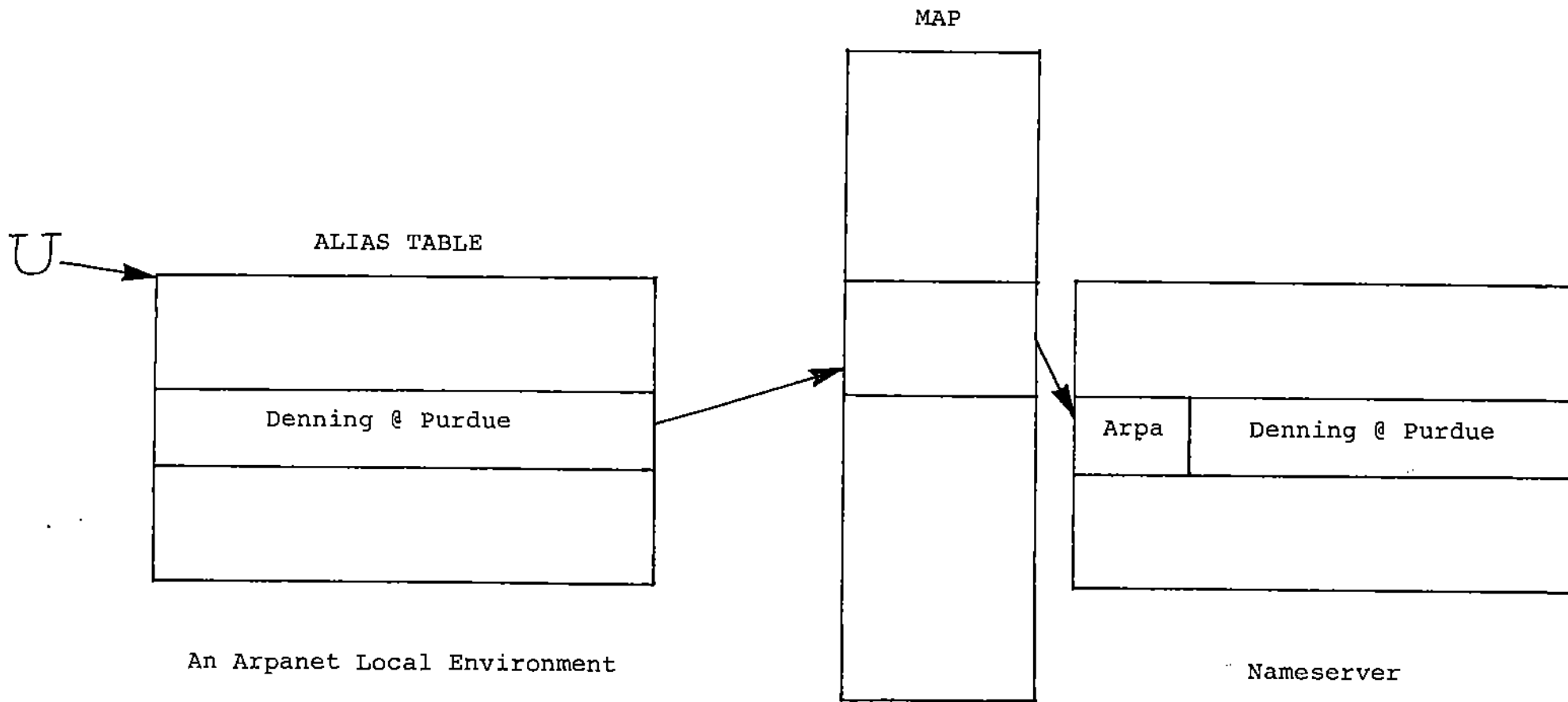


Figure 6: Example of identity map.

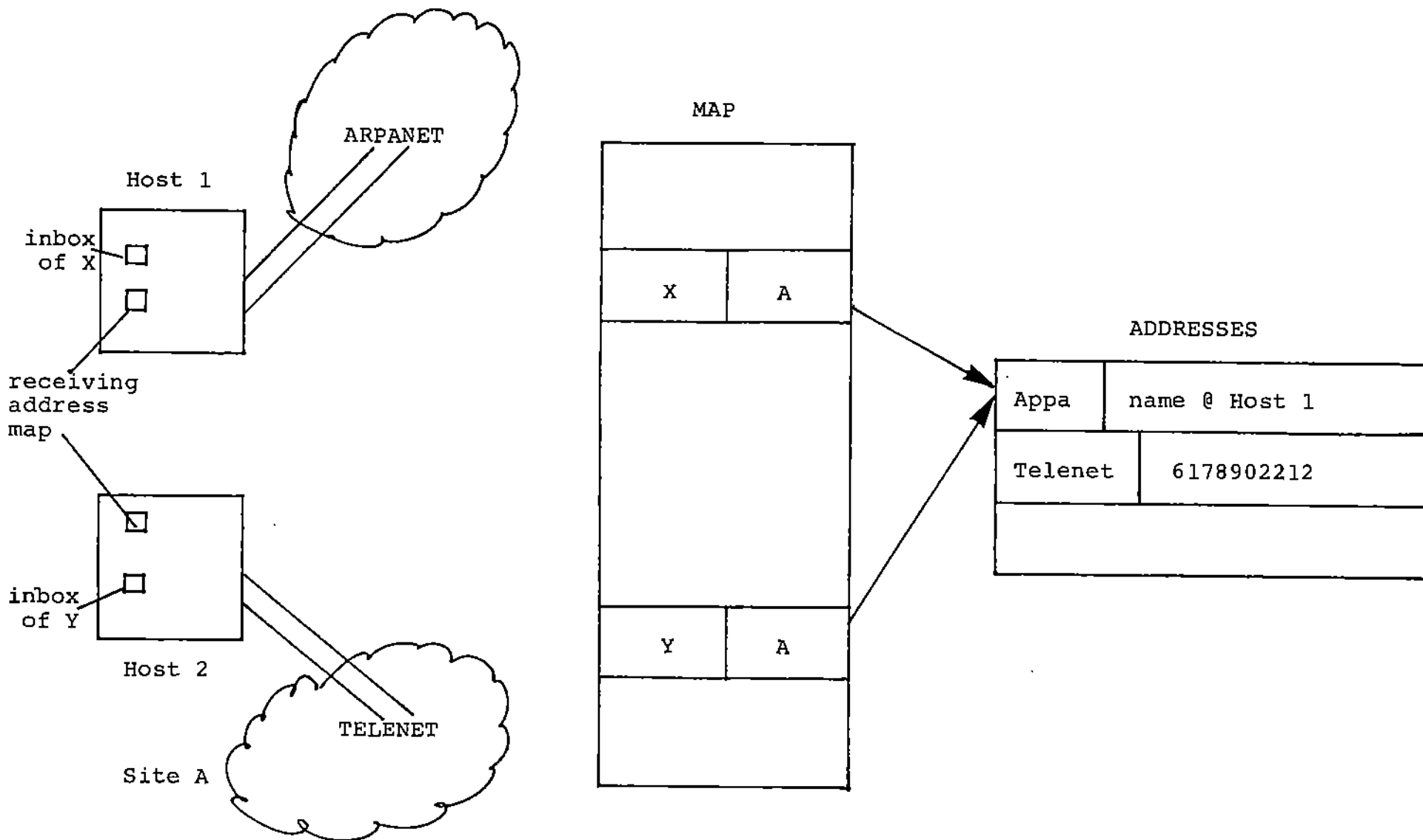


Figure 7: Multihost sites.