# MZnet: Mail Service for Personal Micro-Computer Systems

Einar Stefferud
*President,*
*Network Management Associates*
and
*Visiting Lecturer,*
*in Information and Computer Science*
*University of California at Irvine*

Jerry Sweet
*Department of Information and Computer Science*
*University of California at Irvine*

Terrance Domae
*School of Engineering*
*University of California at Los Angeles*

**Abstract**

Traditional computer mail systems involve a co-resident User Agent (UA) and Mail Transfer System (MTS) on a time-shared host computer which may be connected to other hosts in a network, with new mail posted or delivered directly through co-resident mail-slot programs. To introduce personal micro-computers (PCs) into this environment requires modification of the traditional mail system architecture. To this end, the MZnet project uses a *split-slot* model, placing UA programs on the PCs while leaving MTA programs on a mail relay host which can provide authentication and buffering. The split-slot arrangement might be viewed as a new protocol level which operates somewhere between the currently defined MTS-MTS and UA-UA levels.

# Introduction

Mail systems were born and have grown up on large central time sharing systems, often imbedded in large networks of inter-operating computers with a set of distributed processes automatically transferring mail between users. This is certainly the case with the U.S. Department of Defense (DoD) Advanced Research Projects Agency Network (ARPANET) [1] where much of the original computer network mail systems research and development has taken place. Other mail networks such as the Computer Science Network [2] sponsored by the U.S. National Science Foundation, have also used relatively large shared computers lodged in an institutional setting, though they are often connected together with ordinary dial-up telephone links to form a large geographic network. Another U.S. example is USENET [3] which connects thousands of Unix[1] systems together with informally-supported dial telephone links. Although there have been several attempts, there appear to be no successful mail networks based on small personal computers, such as those that use the CP/M[2] or MS-DOS[3] operating systems.

The accepted architectural model (see Figure 1) for computer network mail (first articulated by the IFIP 6.5 Systems Environment Working Group) involves a User Agent (UA) which posts new mail items through a mail slot [4, 5, 6, 7] to a Mail Transfer Agent (MTA) which delivers posted items to designated UA recipients through corresponding delivery slots. When mail is to be delivered to a UA on another host, it is transferred first to another MTA on the recipient user's host, which in turn puts the mail item through its local delivery slot. In this model, a Mail Transfer System (MTS) may be viewed as a collection of MTAs with network connections among them to provide Mail Transfer Services for a large number of users on different host computers.

Replicating this UA/MTA/MTS model on a personal micro-computer (PC) is not an easy task. Aspects of PCs that make support of this model difficult include limited storage capacities, limited processing capabilities, and the fact that PCs are geared to support a single user rather than several users at once. A PC with limited secondary diskette storage and limited

[1]UNIX is a Trademark of Bell Laboratories, Inc.

[2]CP/M is a Trademark of Digital Research, Inc.

[3]MS-DOS is a Trademark of Microsoft Corporation.

processing capacity (often single-thread) is not well suited to support the full range of automatic interactions between a UA and an MTA, or the necessary interactions between MTAs in an MTS. For example, we do not see any way to certify PC systems for authentication of posted mail. A PC can change its entire character and behavior with insertion of a new program diskette, suggesting that it is the operating system diskettes and their users that must be certified, rather than the computers. Review of certification issues shows that it is not the computer, but its operators and managers that must be certified, and this involves the notions of central management and control. All this is lost in the maze of PCs that we see proliferating on and off our campuses, in and out of our offices and homes.

Thus, we see a need for a new arrangement with the UA separated from its MTA, and using communication protocols to interact with it in ways that resemble MTA-to-MTA interactions. The UA is placed on the PC end, while the more complex tasks performed by the MTA are relegated to the remote host end. The remote MTA must authenticate mail items offered by the PC-based UA, just as it would for a co-located UA, but the task is more difficult because the PC UA is potentially anyone among the public telephone connectable population. This can be handled with password systems, but recognition and identification are not the only services to be provided at the posting slot. Posting also requires some validation of recipient addresses, and validation of the syntax and semantics of certain header fields. Example standards are provided by the U.S. National Bureau of Standards (NBS) and the U.S. DoD ARPANET for the format of mail to be transferred [8, 9, 10].

The new arrangement described in this paper might be called a *split mail slot* in that the UA side of the slot is split away from the MTA side. Although the UA and MTA may be on opposite ends of a telephone connection, they must still act together as a single processing unit to move mail from one to the other, with all that this may entail. This gives rise to a number of new MTA/UA requirements such as error control for service requests, user intervention to select items for delivery, and user postponement or rejection of delivery without triggering failure notices to senders. These are not serious problems when both MTA and UA are programs running on a single host. For example, with both UA and MTA on the same host, unwanted junk mail is simply deleted at low cost, compared to the cost of deletion after a long delivery transmission time. Better that our PC users be able to discard items without delivery transmission.

## Overview of the MZnet Environment

The MZnet project is an undergraduate student effort sponsored within the Information and Computer Science (ICS) Department of the University of California at Irvine (UCI) in Southern California. For the past 2 years, the UCI mail network, known as ZOTnet, has been connected into the Computer Science Network (CSnet) and in 1984, has joined the DoD ARPA Internet with a *Split-Gateway* connection [11] to the University of Southern California Information Sciences Institute (USC-ISI). The MZnet split-slot arrangement may have some similarities to the Split-Slot Internet Gateway at least in name, but the problems and the implementations are quite different.

The UCI ZOTnet environment [13] gives the MZnet project a full-fledged Internet-class mail system as its foundation. The MZnet project objective is to extend this class of mail service to personal computers located in student and faculty residences, offices and laboratories, without waiting for full-blown local area networking to first provide connections. This follows a pattern of making the most of existing facilities to provide a reasonable level of service.

The UCI ZOTnet uses the CSnet-provided MMDF (Multi-channel Memo Distribution Facility) software [12] from the University of Delaware to interconnect two VAX 750 Unix systems with two DEC TOPS-20 systems through a port selector, with dial telephone connection to a CSnet relay [14]. The ZOTnet has since evolved into an ethernet-connected local area network with the aforementioned gateway connection into the DoD Internet. The ZOTnet also connects to USENET with the UUCP protocols, and provides format transformations for mail flowing between protocol domains [15, 16]. Adding to the reach of the ZOTnet with MZnet is a natural part of its evolution[4].

To this point we have set the context of the MZnet project. The remainder of this paper is devoted to relatively technical discussions of implementation of the PC user agent programs and the split-slot UA/MTA interface.

[4]For those who are properly curious about such things, the name "ZOTnet" derives from the cry of the UCI mascot which is the Anteater from the B.C. comic strip, and MZnet is simply a contraction for Micro-ZOTnet.

# The MZnet User Agent: CP/MH

CP/MH is a collection of programs designed to work in conjunction with the Micro ZOTnet (MZnet) as an extension of the UCI ZOTnet. CP/MH programs permit a user of a CP/M 2.2-based microcomputer to send and receive ZOTnet mail messages, as well as to manipulate them locally on floppy disks. The CP/MH programs are written in the C programming language and should be portable to similar operating environments, such as MS-DOS, etc.

CP/MH is based on the UCI version of the Rand MH message handling system [17] for the Unix operating system. The major philosophical differences between CP/MH and typical user agents such as MSG [4] and its descendants are those of modularity and of user interface. In CP/MH (as in MH) the user does not invoke a single monolithic program to deal with mail, but rather invokes individual, non-interactive programs with common knowledge of the way messages are stored. Each program has default behavior which can be modified by using Unix-style command line options at time of invocation or through a user profile. Help messages can also be evoked from CP/MH programs.

## Messages and Folders

The format of a CP/MH message adheres more or less to the syntax described in RFC 822 in which a message consists of headers containing information pertaining to the message source and destination, and the message body, separated from the headers by a blank line. An example of such a message might be:

```
Date: 02 Nov 83 23:04:53 PST (Wed)
To: Toto <dog@Univ-Kansas>
From: The Great And Powerful Oz <Oz@Emerald-City>
Subject: What Be Your Excuse?

What's the matter?  I ask you for a simple thing like
"distribute this to Witch@Oz-West," and you can't do it.
You undergrads will do anything to get out of work!
```

```
--ozzie
```

Following the MH convention, each message is kept in a separate file. Since a message is simply ASCII text, it can be operated upon by non-CP/MH programs (such as text editors, in particular).

Collections of messages are called *folders*. Under CP/MH, folders are represented by several files: an *info* file, containing maintenance information about the folder, and a set of message files with the same name as the info file, but with unique numeric suffixes (*extensions* in CP/M parlance). An example of this naming scheme might be:

`DRAFT` the info file for the `DRAFT` folder

`DRAFT.001` message 1 in the folder

`DRAFT.002` message 2 in the folder

`DRAFT.003` message 3 in the folder

The number of messages that may be stored in a folder is limited primarily by the storage capacity of a floppy disk, but also by the three-digit limit of a CP/M extension.

The info file contains a field named `CURRENT:` specifying the current message number. The current message number signifies the default message operated upon by CP/MH commands using a particular folder. The current message number may be modified by some commands. An example of the contents of the info file `DRAFT` might be

```
CURRENT: 3
```

This indicates that the file `DRAFT.003` would be operated upon when default conditions apply (i.e. when no message number is explicitly given to a CP/MH command).

Possible future uses for the info file include named message sequences (a set of messages to which commands may be applied as a whole) and user profile information for application to particular folders (there is presently a single user profile, described shortly).

A floppy diskette may contain more than one folder, but folders do not extend over more than one floppy diskette; therefore two different diskettes may contain folders with the same name.

## CP/MH Commands

Commands operating on messages can be divided into several general categories:

**Transporting:** sending, receiving

**Viewing:** selecting for display, showing header summaries

**Creating:** composing, replying, forwarding

**Archiving:** categorizing, refiling, deleting, sorting

The architecture of CP/MH permits the simulation of some of these categories using standard CP/M commands when CP/MH, in its present primitive state, does not cover them.

A minimal functionality is presently provided by the following four commands:

**COMP** composes mail items: creates a file containing header information taken from a standard or user-specified template. This newly-created file may be edited to fill in the header fields and body.

**REPL** replies to mail items: creates a file containing header information appropriate for answering a given mail item. This newly-created file may be edited to change header fields and fill in the body.

**SEND** sends mail items: posts selected items through the split-slot from a draft folder.

**INC** receives mail items: takes delivery of selected items across the split-slot, incorporating them into a mailbox folder.

These commands, with a few enhancements and modifications appropriate to the CP/M environment, are functionally almost identical to their Unix MH counterparts.

CP/MH commands are invoked like any other CP/M commands such as ED, PIP, or DIR. Command line options are generally preceded by a dash (e.g. `-editor A:ED`), and may be abbreviated. Folder names are preceded

by a plus (e.g. `+B:DRAFT`). Messages are identified by numbers or by the special names `first`, `last`, `current`, `next`, and `previous`.

An example of use of a CP/MH command is:

```
comp -edit a:ed -use last +b:draft -log
```

This particular example will edit the last-composed message (the `-use last` option) in the folder `DRAFT` on disk drive `B:` (the `+b:draft` option), using the standard CP/M editor ED on disk drive `A:` (the `-edit a:ed` option), and prompting the user when it is appropriate to change disks (the `-log` option).

All CP/MH commands have a `-help` option which displays all available options for the particular command invoked. Another common option is `-log` which permits the user to change (*relog*) diskettes after invoking a command, for purposes of selecting diskettes with message folders or with editor programs. This is particularly useful on single-drive systems or on systems with diskettes of low storage capacity.

## The Profile

If there are options commonly used with a particular CP/MH command, they may be entered in the user profile contained in the file called (naturally enough) `PROFILE`, which must exist on the same diskette on which CP/MH commands reside and from which the commands are invoked. A profile entry consists of a program name followed by a colon and the options to be used with that program, for example:

```
comp:  -editor A:VEDIT +B:outbox -log
repl:  -editor A:VEDIT -log
send:  +B:outbox
inc:  +B:inbox -log
```

Individual profile components are overridden by options given at the time of invocation (e.g. `-noedit` given on the command line will override the `-editor` profile component for a particular command).

# The MZnet Split-Slot Mail Transfer System

The MZnet split-slot software implements a peer-to-peer communication protocol between a time-sharing host's MTA and a personal micro-computer (PC) UA. This MZnet protocol extends the UA/MTA/UA model of computer-based message systems (CBMS) to provide a split gateway function between individual PCs and the ZOTnet similar to the UCI ICS split Internet gateway described previously (see Figure 2).

## The Structure of the Split-Slot

The MZnet Split Gateway consists of three distributed processing components:

- A PC running a UA (in MZnet, CP/MH) acting as the mail server.
- A mini/mainframe host running a full MTA (MMDF in MZnet) providing mail relay services.
- A communication protocol (a modified version of MMDF PhoneNet) to connect the two ends of the split-slot.

Although this combination may not be unique, the method by which the MZnet split-slot bonds these parts together uniquely deals with the problems of remote user agents. In addition to overcoming limited storage and processing capacities, remote user agents must deal with noisy modem lines, mail software certification, and mail system security problems. The MZnet architecture appears to solve these problems with a clean mail interface for PCs.

## The MZnet Mail Server

The split-slot mail server consists of a set of *command packet* programs run from the PC. These programs simply present commands through the PhoneNet communication protocol to the mail relay slave program on the host. Some basic commands are:

**PostMail** posts mail drafts to MTA

**GetMail** accepts mail from MTA

**RemoteScan** displays information about waiting mail

**Quit** drops connection between PC and Host

Each command has the form:

Command Request
Data Transmission
Command Termination

For example, the PostMail command is a small program that:

- initiates a command with the Mail Slave by sending the command name (PostMail) encoded within a PhoneNet packet;
- sends a series of PhoneNet packets that contain pieces of the mail item to be posted;
- finally sends a command termination signal to end the transaction without terminating the connection between host and PC.

## The MZnet Channel To MMDF

The MZnet Channel runs on the MTA host under the University of Delaware's MMDF (Version 1) and is responsible for both delivery of received mail to MZnet users, and posting of MZnet user-originated mail. The MMDF MZnet channel maintains a unique message queue for each registered MZnet user. As new mail items arrive, they are posted to the appropriate queues, where MZnet holds the mail items for pickup by their registered recipients.

To send or receive mail, the MZnet user must attach to the host, log into the public MZnet account, and identify (authenticate) himself. During the MZnet session with the host, the user has access only to that restricted set of functions provided by the MZnet split gateway protocol: he may request delivery of queued mail with GetMail, or post new mail with PostMail. Prior to taking delivery of queued mail, a survey of waiting mail also may be

requested with RemoteScan to obtain message size information (among other data) to allow intelligent disposition of mail in the queue.

Hidden within these activities are issues of security and certification. To certify and establish the identity of the user, a second password is requested after logging into the public MZnet account. This certification procedure allows MZnet to certify the source of originated mail. A relatively secure environment is provided by MZnet, as it is the only interface to the host permitted to MZnet users (once beyond the public login procedure), and it offers only the severely restricted set of PhoneNet-encoded commands. Aside from security issues, using a single account to handle all MZnet users reduces demands on system resources.

## The MZnet-PhoneNet Protocol

A unique facet of the MZnet system derives from the PhoneNet File Transfer Protocol (FTP). PhoneNet FTP is a simple error-checked packet protocol which transfers ASCII plaintext. PhoneNet encodes any non-plaintext character (or any other character "forbidden" by the idiosyncrasies of the communicating systems) by mapping it onto an "accepted" character set. The accepted character set mapping is determined by a "negotiating" session between the two systems at the start of the PhoneNet session.

MZnet transfers all information (both commands and data) in PhoneNet packets to obtain error control. The MZnet-PhoneNet command FTP tolerates noise with a high degree of success, and in effect, connects both ends of the Split Slot together with a reliable set of virtual wires.

## MZnet Session Example

Here, a typical MZnet session is presented, with the UA commands issued from the PC side of the connection printed in a `typewriter` typeface, and the responses from the host side printed in an *italic* typeface. PhoneNet interactions are indented. The initial connection to the host is accomplished with the `term` program, which provides a simple terminal emulation function. The prompt of the PC for a UA command is "A⟩". Note that passwords are never echoed by the host system.

A⟩ `term`

*login:* `mznet`
*password:*
*MZ-Password:*
PhoneNet packet negotiation
*Connected.*
exit terminal mode
A⟩ `send cur`
PostMail command
message text packet transmission
command terminator
A⟩ `quit`
Quit command
*Disconnecting.*

# Conclusions

The main conclusions of this paper are that small personal computer systems with dial-up phone connections constrain User Agent systems design in ways that require use of a *split-slot* interface between the UA and its supporting Mail Transfer Agent (MTA), and that this interface will best provide the required services if it has error controlled command and data transfer facilities, with interactive behavior.

It is also believed that a good design for the small PC UA is based on a very modular architecture, such as the Rand MH system, which has been used as a pattern for the MZnet UA.

By bringing these concepts together, we expect MZnet to provide reliable UA/MTA service to a distributed set of small personal computers, to match the quality of service that is normally only available from larger mainframe host systems with co-resident UA/MTA pairs.

# References

[1] SRI-NIC, ARPANET Directory, Network Information Center, SRI International, Menlo Park, California (November 1980).

[2] Comer, D., A Computer Science Research Network CSNET: A History and Status Report, Communications of the ACM, volume 26, number 10 (October 1983) 747-753.

[3] Emerson, S. L., USENET: A Bulletin Board for Unix Users. BYTE, volume 8, number 10 (October 1983) 219-236.

[4] Vittal, J., MSG: A Simple Message System, in: Uhlig (editor), Proceedings of the IFIP TC-6 International Symposium on Computer Message Systems (North-Holland, April 1981).

[5] Deutsch, D., Design of a Message Format Standard, in: Uhlig (editor), Proceedings of the IFIP TC-6 International Symposium on Computer Message Systems (North-Holland, April 1981).

[6] v.Bochmann, G. and Pickens, J. R., A Methodology for the Specification of a Message Transport System, in: Uhlig (editor), Proceedings of the IFIP TC-6 International Symposium on Computer Message Systems (North-Holland, April 1981).

[7] Kerr, I. H., Interconnection of Electronic Mail Systems, in: Uhlig (editor), Proceedings of the IFIP TC-6 International Symposium on Computer Message Systems (North-Holland, April 1981).

[8] Crocker, D., Standard for the Format of ARPA Internet Text Messages (RFC 822) Network Information Center, SRI International, Menlo Park, California (August 1982).

[9] NBS, Message Format for Computer-Based Message Systems, U.S. National Bureau of Standards FIPS Publication 98 (March 1983).

[10] CCITT Study Group VII/5, Draft Recommendation X.MHS1: Message Handling Systems: System Model—Service Elements (version 2), Technical Report, International Telegraph and Telephone Consultative Committee (CCITT) (December 1982).

[11] Rose, M., Low Tech Connections into the ARPA Internet: The Raw-Packet Split-Gateway, University of California Irvine Techical Report number 216 (February 1984).

[12] Crocker, D., Szurkowski, E., Farber, D. J., An Internet Memo Distribution Facility—MMDF, Proceedings of the Sixth IEEE Data Communications Symposium (November 1979).

[13] Rose, M., The ZOTnet—A Local Area Mailing Network, University of California Irvine Technical Report number 200 (January 1983).

[14] CSNET-CIC, Focus on the University of California, Irvine, CSNET News 2, Bolt, Beranek, and Newman, Cambridge, Massachusetts (October 1983).

[15] Rose, M., Achieving Interoperability Between Two Domains—Connecting the ZOTnet and UUCP Computer Mail Networks, University of California Irvine Technical Report number 201 (January 1983).

[16] Rose, M., Proposed Standard for Message Munging (RFC 886), Network Information Center, SRI International, Menlo Park, California (December 1983).

[17] Borden, B. S., Gaines, R. S., and Shapiro, N.Z., The Rand MH Message Handling System: User's Manual (Rand Corporation, March 1983).

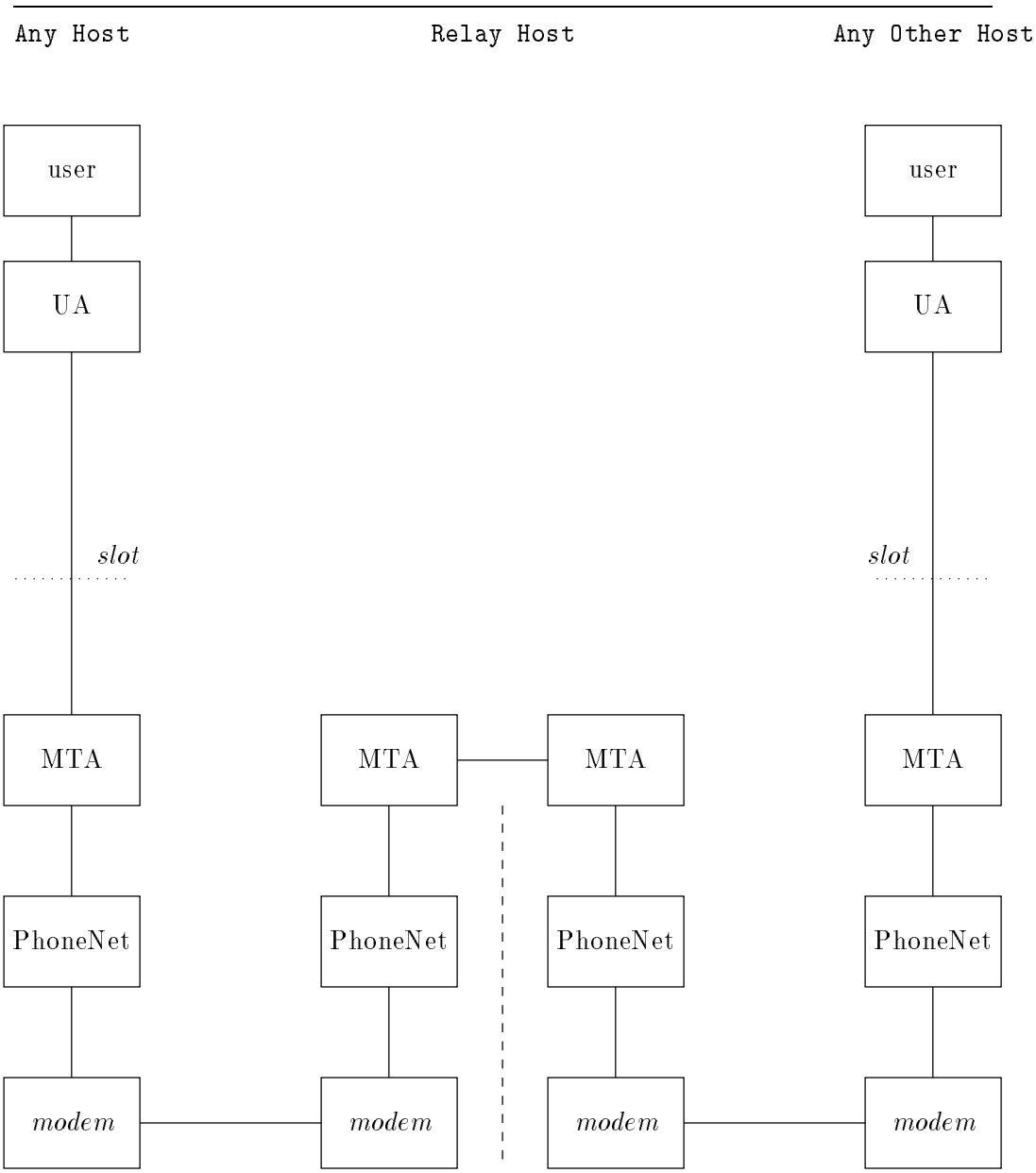


Figure 1: The MHS Model

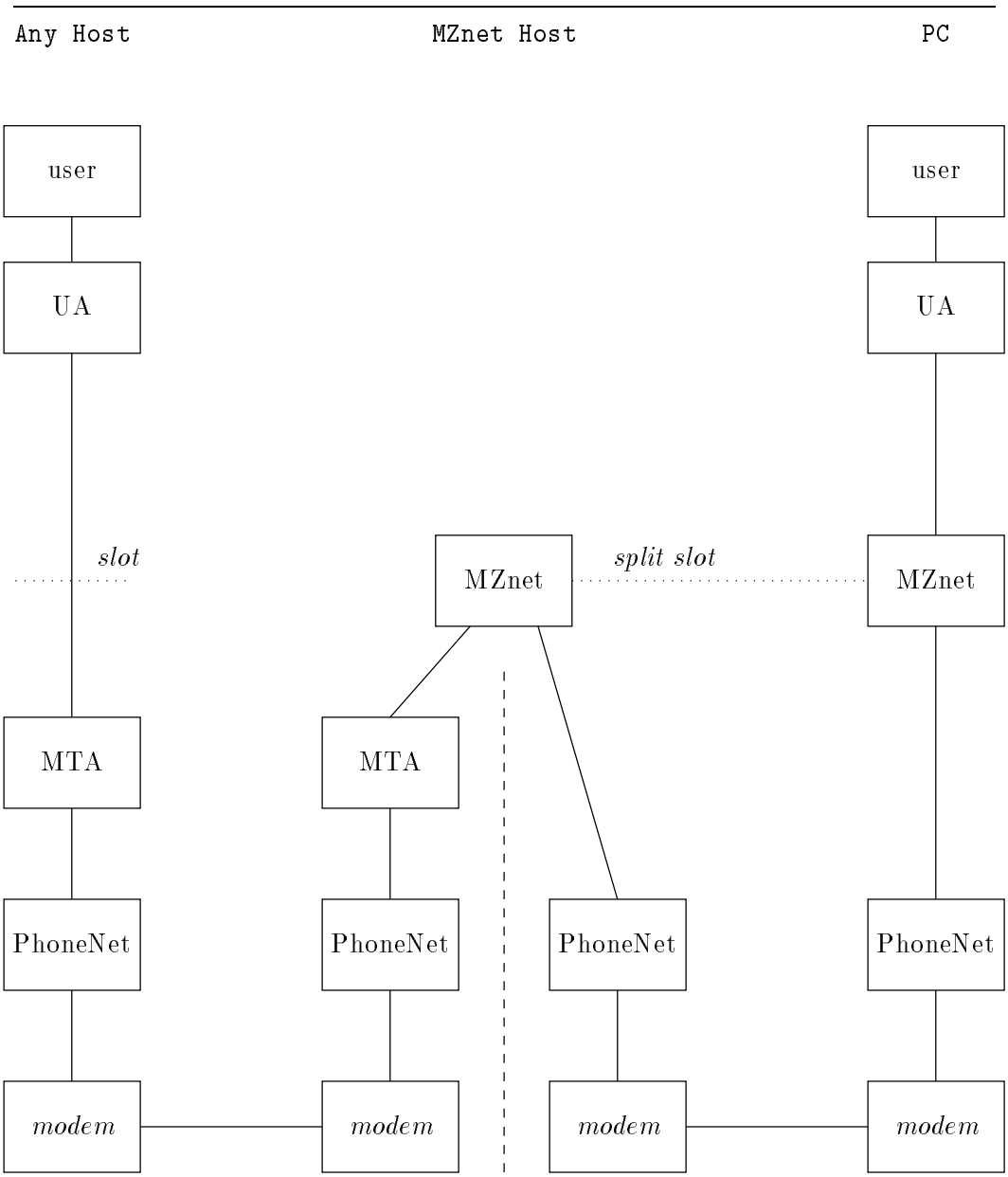


Figure 2: The Split-Slot Model