

LispCourse #19: Living in the Network World

Basic Concepts

Advantages of Networks

Interlisp-D runs perfectly well in a stand-alone, single machine world.

But, the full power of Interlisp-D comes when your D-machine is attached to a network and can communicate to the other machines on the network.

Networked machines have two basic advantages:

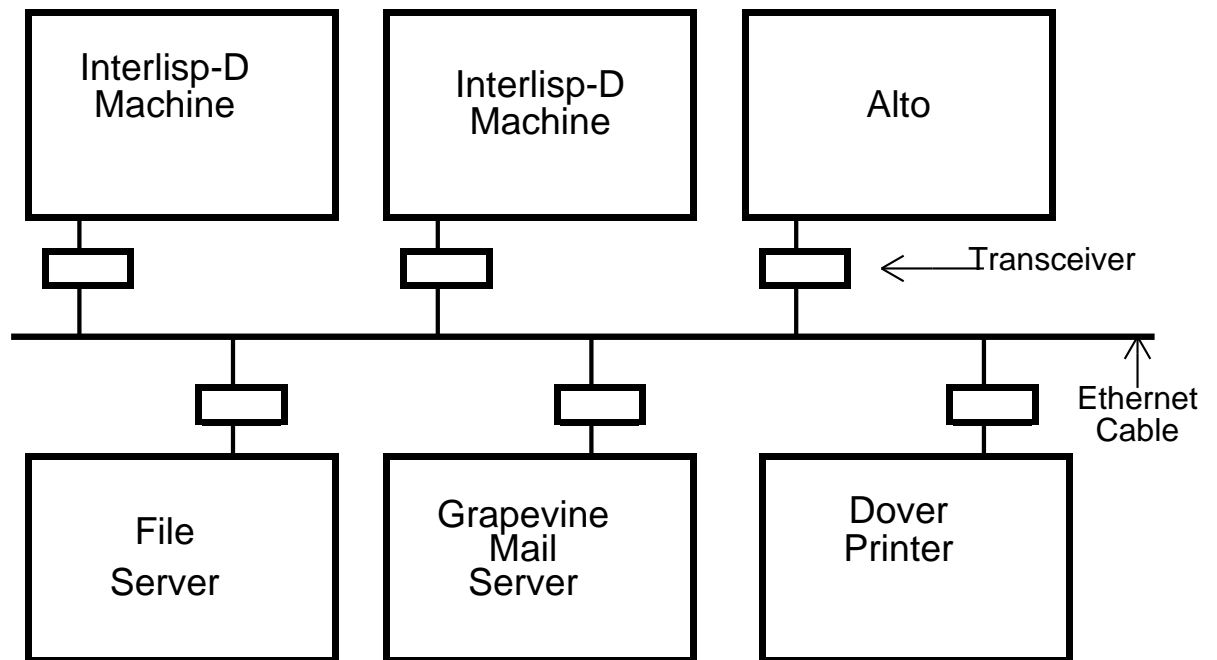
- 1) *Communication* ž you can communicate with other users on other machines on the network. Examples: electronic mail and sharing information by storing a common file on a file server.
- 2) *Sharing of expensive resources* ž many machines can share expensive resources such as printers, disk drives, etc. Without a network, every machine would need its own printer, its own large disk, its own tape drive, etc.

Basic Network Concepts ž Servers and Clients on a Local Network

Xerox's brand of network is called an Ethernet. At the level of this discussion, the Ethernet is pretty much like most of the Local-area networks one can buy to connect together groups of computers.

The Ethernet is simply a coaxial (i.e., cable television) cable. Two or machines can "tap" onto this cable with a special piece of hardware called a *transceiver*.

The various computers on the Ethernet can communicate to each other by sending messages over this cable.



In principle, any two machines on this network can communicate with each other. For example, the two Interlisp-D machines could communicate with each other directly.

In practice, machines on an Ethernet are usually separated into two classes: **servers** and **clients**.

Servers – machines that exist to perform one or more specialized services such as storing files, printing documents, processing mail, etc. These machines sit on the network and wait for a client machine to request some service. Once the request arrives, they perform the service.

Clients – the ordinary user machines. Programs (e.g., Interlisp-D) that the user runs on these machines occasionally need some specialized service (e.g., a file printed, a file stored, mail sent or retrieved) that they request from a server somewhere on the network.

Most communication on a network takes place between clients and servers.

Communication between servers and servers also take places as different servers need to coordinate their services. For example, when I send mail to you, my mail server communicates with your mail server in the process of transferring the mail.

Some typical servers:

Boot Servers ž provide programs necessary to boot machines over the network. For example, provide Othello during 3-boot of the DLion or provide the NetExec while net-booting a Dorado/Dolphin.

Name Servers ž provide translations from names to locations on the network. For example, tells Interlisp-D that PHYLUM is located on network number 6, and is machine number 255 on that network.

Authentication Servers ž provide name and password checking. When a user wants to log into any machine on the network, that machine okays the user's name and password with an authentication server.

File Servers ž provide space to store files.

Mail Servers ž provide electronic mail delivery, distribution lists, etc.

Print Servers ž computer attached to a printer; queues and prints documents.

Specialized Servers ž e.g., dictionary server, dial-in servers, tape servers, etc.

NOTE: A server is a program, i.e., a piece of software. On any given machine, there may be several servers (i.e., programs) running at once. The servers are logically separate, though physically dependent. For example, it is standard to run the Boot server and the Name server on the same machine on a network.

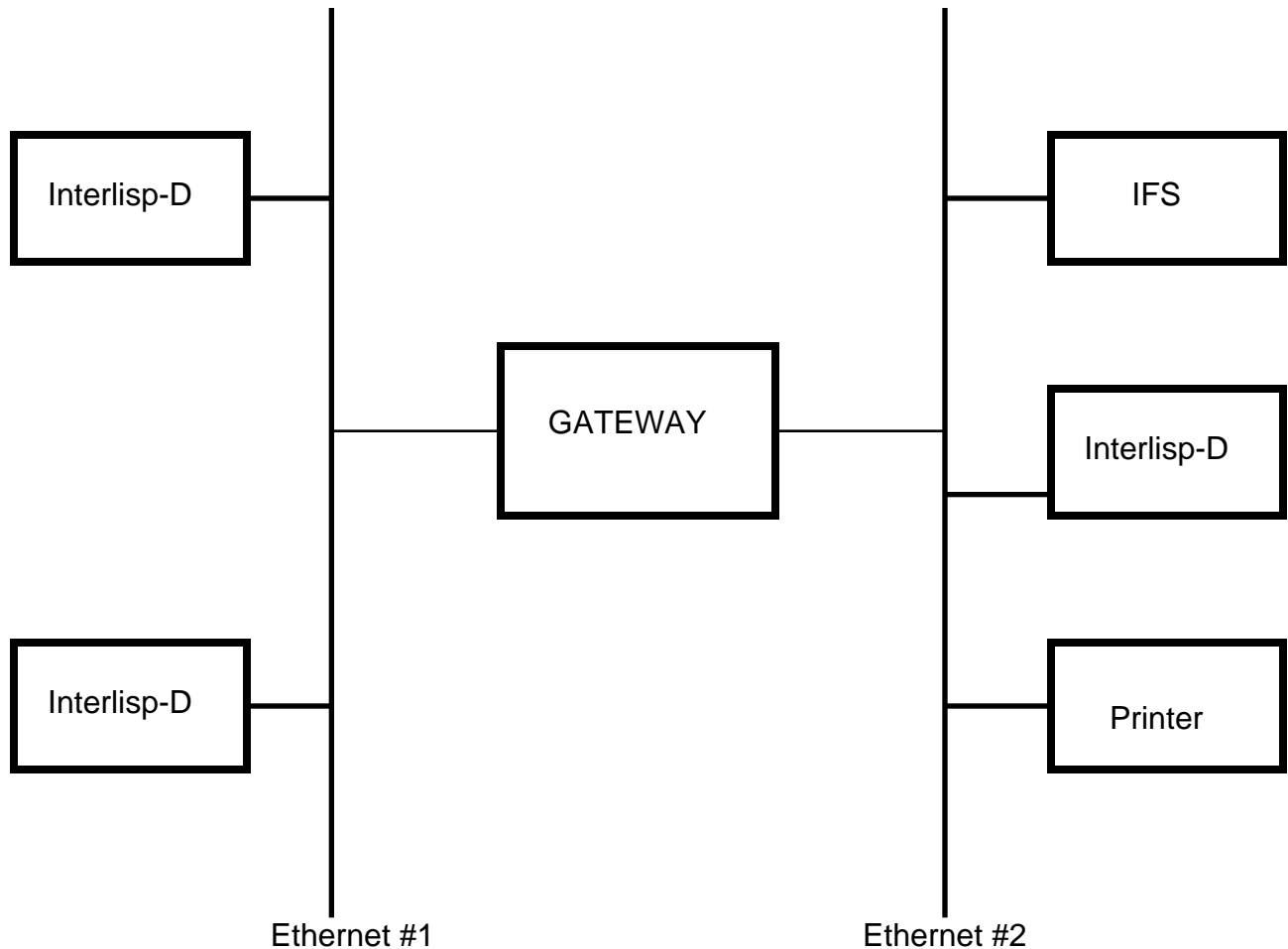
Moreover, on PUP networks (see below), the same Grapevine "server" (read, machine) provides both mail and authentication service.

The Internet: Connecting Together Many Local Networks

A single local network can handle only a limited number of machines that are located within a relatively small distance of each other.

Typically, local Ethernets are connected together to form a larger "network of networks" called an *internet*, as in "*The Xerox Internet*".

Local networks are connected together using *gateways*.



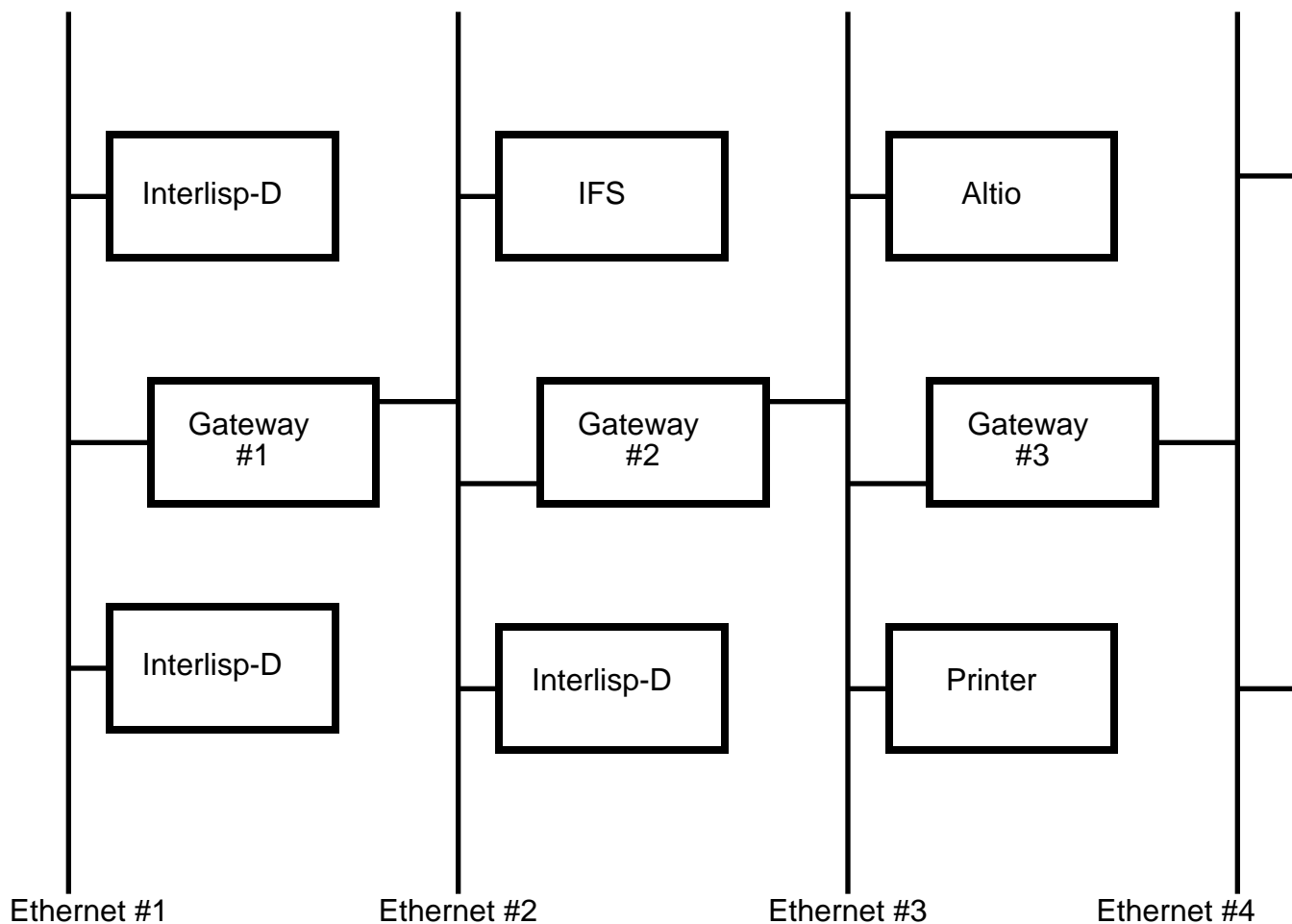
A **gateway** is a computer that is connected to two or more networks. If a machine on one network wants to send a message to a machine on another network, it sends the message to a gateway on its network. The gateway takes the message off of the network of the originating machine and puts it on the network of the destination machine.

In short a gateway transfers message from one local network to another local network. If your machine is on one network and your file server is on another

network, when the gateway goes down you are effectively cut off from your file server.

The Xerox Internet consists of many, many (100??) local Ethernets connected together by gateways. There are maps of this network on ????

In the Xerox Internet not all of the Ethernets can talk directly through a single gateway to all of the other Ethernet. A message may have to pass through several networks and several gateways to get from one machine to another. For example, to get from an Interlisp-D machine on Ethernet#1 to a Printer on Ethernet#3, the message would have to pass from Ethernet#1 through Gateway#1, Ethernet#2, and Gateway#2 to Ethernet#3.



Gateways also do **routing**, i.e. they help figure out what sequence of Ethernets and Gateways will get a message from its source to its destination.

In particular, the source machine sends the message to a gateway on its network. This first gateway transfers the message to a second network and sends the message on to some other gateway on the second network that is also connected to a network that is closer to the destination. This gateway passes the message on to an even closer gateway. This process repeats until the message reaches its destination.

The point is that no gateway needs to "know" the entire route for the message. Each gateway only needs to know what networks are likely to be "closer" to the destination machine than the gateway that gave it the message.

You don't need to know much about gateways. Internetwork transfer and routing is all done for you (the Interlisp-D user) automatically.

But: gateways do go down sometimes and sometimes they get overworked and slow down. In this case, message traffic on the Internet can slow down or stop altogether for some machines.

Example:

The ISL Dorados are on Ethernet #6.

Phylum is also on Ethernet #6.

The ISL DLions are on Ethernet #204.

NewWing is a gateway between Ethernets #6 and #204.

If NewWing goes down or slows down, then the Dorados will be able to get at Phylum with no problem, but the DLion will have problems because their message are not being transfered from net #204 to net #6.

Note: In general the machines that run gateways are the same machines that run the Name server and the Boot server on the Ethernet, although all three are logically separate services that could be run on three different machines.

Physical Ethernets ž 3Mb Experimental versus 10Mb Product

There are two different kinds of physical Ethernets: the 3Mb Experimental Ethernet and the 10Mb Product Ethernet. The two are distinguished by the transceiver hardware and the type of coaxial cable used. They also transmit at different speeds - 3 million bits per second versus 10 million bits per second.

Most machines have the hardware to support either the 3Mb Ethernet or the 10Mb Ethernet, but not both.

In general DLions and all product equipment uses the 10Mb Product Ethernet. The Altos (including all IFSs and many mail servers), Dolphins, and Dorados use the 3Mb Experimental Ethernet (i.e., the original PARC invention).

There are NO logical or functional differences between the two Ethernets. In the Xerox Internet, 3Mb and 10Mb Ethernets are mixed together. There are gateways containing both 3Mb and 10Mb Ethernet hardware that transfer messages between 3Mb networks and 10Mb networks.

The bottom line: if you have both a Dorado and a Dlion, they can't be on the same network. You'll need two separate Ethernets running through your office, one 10 and one 3. The two machines can speak to each other, but only through a gateway that connects 3 and 10 Mb networks. (Special note: you can actually but Dorados with 10Mb hardware, but we don't have any in ISL.)

Communicating on an Ethernet ž Packets and Protocols (PUP versus NS)

Packets

The Ethernet is a packet network: each message between two machines is broken down into short **packets** of information. These packets are then sent out over the Ethernet one after the other until the entire message has been transmitted.

The transmission of a packet between two machines is actually a "conversation" between two the two machines. The source machine sends out a packet. The source machine then waits before sending out the next packet until it recieves some sort of acknowledgement. If the destination machine receives the packet correctly, then it sends an "okay" acknowledgment packet to the source machine. If it receives the packet with some sort of error, it sends back an error acknowledgment packet. If

the source receives an error acknowledgment , it resends the packet, otherwise it goes on to the next packet.

Several "conversations" can be going on at once on an Ethernet. While one computer is preparing the next packet to be sent, another computer can be actually sending its packet on the cable. Thus, the various machines on an Ethernet share (technically, *multiplex*) the available time on the Ethernet.

Protocols (PUP versus NS (versus TCP/IP))

For two machines to carry on a conversation over a network, they must see eye-to-eye on a few matters. In particular, they must agree on the format of each packet and they must agree on the exact meaning of each of the message transmitted.

There are standards, called **protocols**, that specify these two things.

Level One Protocols

The standards for the format of each packet are called **level-one protocols**.

There are *three* different level-one protocols in use within Xerox:

- the PUP protocol ž the original PARC protocol

- the NS protocol ž the product protocol

- the TCP/IP protocol ž the Arpanet and university-favorite protocol

Each of these protocols specifies a slightly different format for each packet of information sent across the network. The major difference is in how you specify the network and machine numbers.

Most systems can speak (and understand) only one of the protocols. For example, Star and the NS file servers speak only NS. Cedar, the IFSs and the Grapevine speak only PUP protocols. Only the Vax speaks TCP/IP.

This means that a Star cannot speak to an IFS or to a Grapevine mail server because the two cannot agree on the format of the Ethernet packets.

Interlisp-D, however, can speak all three level-one protocols (TCP/IP will be available in Intermezzo). Therefore, Interlisp-D can exchange packets with Star and the NS file servers AND with the IFSs and Grapevine mail servers AND with the Vaxes running TCP/IP.

[Note: Some gateways handle only NS packets. I have been stuck in the situation where there are two machines that both speak PUP and were physically connected together by a sequence of gatewayst. But the two machines could not communicate because one of the gateways connecting the two machines could not handle PUP packets. Ugh!!!]

Higher Level Protocols

Agreeing on a packet format is not enough. To communicate, two machines must agree on the set of possible messages and the exact meaning of these messages.

For example, if one machine wants to ship a file to another machine, the two machines must agree on the exact set of messages to exchange in order to get this done: how to specify the file name on the destination machine, how to encode the data in the file, how to signal errors in the received packets, how to notify the destination machine that the file transmission is complete, and so on.

The standards for the exchange of these "task-oriented" messages are known as **higher-level protocols**.

There are a multitude of higher level protocols:

file transfer protocols

mail protocols

telnet (CHAT) protocols

leaf (page level file access) protocols

and many, many more

Each of these protocols specifies the set of messages that need to be interchanged between two machines to carry out some specific task like transferring a file, delivering mail, opening a CHAT connection, etc.

Example:

An file transfer protocol might have the following messages:

Here's the file name to call the file: *Name*

Here's the next 100 bytes of the file: *Data*

Here's bytes N thru M of the file: *Data*

File transmission is done.

Bytes N thru M received okay (in error).

File name not allowed.

...

The bottom line: if two machines speak the same task-oriented protocol, they can effectively communicate in carryoing out a task, e.g., in transfering a file. If they don't speak the same task-oriented protocol, there is no way they could communicate well enough to carry out the task.

The PUP World versus the NS World

In principle, the same higher-level protocol could be carried out using either NS or PUP level-one packets.

In practice, the PUP-based higher-level protocols are entirely different from the NS-based higher level protocols. Basically, when PUP was redesigned into NS, all of the PUP-based higher-level protocols were totally redesigned as well.

For example, the file transfer protocol for PUP-based file servers (i.e., FTP protocol) is entirely

different from the file transfer protocol for NS file servers (the NS filing protocol).

Star can't speak to an IFS because it can't produce PUP packets AND it doesn't support the FTP protocol.

Interlisp-D exists in both worlds (in fact in three worlds because there are also TCP/IP higher level protocols). From Interlisp-D you can access both IFSs (using the FTP protocol) and the NS file servers (using the NS filing protocol).

The PUP World

Addresses

In the PUP world, every machine is uniquely identified (i.e., addressed) by an 8-bit (0 to 255) network number and an 8-bit machine number. The complete address of the machine is the network number and the machine number, each followed by a "#".

Examples:

Phylum's address is 6#225#

My Dorado is 6#60#

The pool DLion in ISL is 204#36#

QUAKE is 6#357#

EXPRESSO is 64#154#

Certain programs require you to specify a destination machine by its PUP address; TeleRaid, for example.

Interlisp-D also allows you to specify a PUP address in place of a device name for any device reference. For example, instead of (*CHAT 'PHYLUM*), you can use (*CHAT '6#225#*). Instead of *DIR {PHYLUM}<HALASZ>*, you can use *DIR {6#225#}<halasz>*. These addresses can come in handy when the name server is down.

Name Service

Most machines in the PUP world are given a literal for a name; this is true of all servers and most client machines. Thus the name of my machine is Halasz, the name of machine 6#225 is Phylum, etc. Note: some DLions in ISL do not have name because they ran out of space in the name tables.

The name servers running on the PUP-gateways keep a table that translates these names into PUP addresses. Whenever you refer to a machine by name [as in (CHAT 'PHLYUM)], Interlisp-D consults the name server for the address corresponding to that name.

In order to get to a particular type of server in the PUP-world, you have to know the name or PUP address of the particular machine or the machine running the server.

The PUP name servers cannot answer interesting questions like *"What machines are on net 36"* or *"What IFSs do you know about?"*.

In Interlisp:

(ETHERHOSTNAME *PupAddress*) ž returns the name of the machine whose address is *PupAddress*. (ETHERHOSTNAME '6#225#) returns *Phylum*.

(PORTSTRING (ETHERHOSTNUMBER *Name*)) ž returns the PUP address of the machine specified by *Name*. (PORTSTRING (ETHERHOSTNUMBER 'Halasz)) returns *6#60#*.

Authentication Service

In the PUP world, users are registered using the Grapevine mail system.

All users are registered on a Grapevine mail server in their location (called a *registry*). A user's Grapevine name is usually his last name followed by the registry where he works. For example, my Grapevine name is *Halasz.pa* since *.pa* is the Palo Alto registry. Other registries are:

.wbst ž Webster, N.Y.
 .pasa ž Pasadena
 .es ž El Segundo
 .sthq ž Stamford Headquarters
 .henr ž Henrietta, N.Y.

.dlos ž Dallas

???

When a user logs onto any machine on the network, that machine asks the nearest Grapevine server to authenticate the user and her password.

Because all Grapevine mail servers are in constant communication, you can log into a machine anywhere on the Xerox Internet. That machine will query the Grapevine server nearest to it, which in turn will query your Grapevine server here at PARC. If you typed in the right password, the PARC Grapevine will say okay to the remote Grapevine server which will say okay to the machine you are logging in to.

The Grapevine is a very limited database of all of the users on the Xerox PUP-based Internet. The Grapevine system maintains a list of all the user's on the Xerox Internet. However, it keeps only the user's name and no other interesting information such as office number or phone number or position in the company.

Moreover, you can't even access the list of users in any very interesting way. For example, you can't ask "what users are there in Palo Alto whose last name is Jackson?" or even "Who works at PARC?".

(This contrasts with the NS world, where more interesting queries are possible).

Mail Service

Mail in the PUP world is also handled by the Grapevine system. The recipient of a mail message is specified by his or her Grapevine user name (e.g., Halasz.pa).

When you send mail in Interlisp-D, Lafite sends the mail to the nearest Grapevine server which then distributes the mail to the Grapevine servers of each of the recipients of the mail. The Grapevine system does the work of figuring out the relevant Grapevine server for each of the recipients.

When you retrieve mail, Lafite queries the Grapevine servers for your registry to see if there is any mail waiting for you. If so, it copies the mail from the Grapevine server to your Lafite mailbox.

In addition to delivering and storing mail, the Grapevine provides for the maintenance of a set of distribution lists. Each distribution list consists of a name ending in an "^" (e.g., LispUsers^), a registry (as in LispUsers^.pa), and a list of members of that list. Sending mail addressed to a distribution list will cause the mail to be sent to all of the members of the distribution list.

You can log into the Grapevine and alter various aspects of distribution lists. For example, you can add or remove yourself from a distribution list, see who is on a distribution list, etc.

To do this from Interlisp, LOAD the library package *{eris}<lisp>harmony>library>maintain.dcom*. When the LOAD is finished, the function call (*MAINTAIN*) will log you into the Grapevine and start up a Grapevine "exec" in your TTY window.

From the Grapevine "exec" you can execute a number of commands to query or change distribution lists.

Example:

When you are finished with your interaction with Grapevine, type "Quit" to the exec. This will return you to the Lisp Exec in the TTY window.

For full documentation of the Grapevine "exec", see the document *{indigo}<laurel>maintain.press*.

File Service, Print Service

In the PUP-world, file service is provided by the oft-discussed IFSs.

In the PUP-world, print service is provided by the Dover printers such as Quake and Espresso and the full-press printers such as Jedi and RockNRoll.

Before a file can be printed, it must be translated into a format that the printer can understand. Printers in the PUP-world print PRESS format files, in contrast the NS-worlrd printers which print InterPress format files. While these file formats have nothing to do with PUP and NS in principle, in practice Press versus Interpress follows the PUP versus NS distinction.

Still to Come on Networks

The NS World

Cross-overs between PUP and NS worlds