Researches in Network Developement of JUNET

Jun Murai[1]

Computer Center
University of Tokyo
2-11-16, Yayoi, Bunkyo
Tokyo, 113 JAPAN

Akira Kato[2]

Department of Computer Science
Faculty of Engineering
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro
Tokyo, 152 JAPAN

Abstract

JUNET was developed in order to provide a testing environment for studies of computer networking and distributed processing by connecting a large number of computers and by providing actual services for the users. Research interests in development of the network have been focused on resource name managing, Japanese character handling and communication technologies.

For the name managements, the hierarchical domain concept is employed to construct a name space for the network, and a mechanism for text message exchange is implemented using the concept. An environment for text message exchange using Japanese characters is achieved as results of general discussions to handle 16-bit Kanji codes in computers. Efficient data transmissions with the high speed modems are achieved by a new UUCP protocol and the dial-up IP link mechanism developed with a tty driver which provides host-to-modem flow control mechanism.

As the result of researches described above, JUNET currently connects various types of organizations relating computer science which are 87 organizations with more than 250 computers in number. It connects universities and major research laboratories in Japan, and the protocols currently used are TCP/IP over leased lines as well as dial-up lines, and UUCP over dial-up lines. The services such as electronic mail and network news have been provided since the network was started, and special technologies for Japanese character handling, name servers, and multimedia mail supports have been developed.

In this paper, the current status of JUNET and its research results in development of the network are described.

[1] jun%u-tokyo.junet@relay.cs.net
[2] kato%cs.titech.junet@relay.cs.net

1 Introduction

Since JUNET is the first and the only existing national wide computer network in Japan where user can exchange electronic mails and network news domestically and internationally, most of the experimental studies about wide-area computer networks, from physical communication technology to application design, have been done using the network. The purpose in the very first stage of the network, thus, was providing actual network services to researchers and put Japanese communities into world academic networks. And then, we started to work to solve problems existing on the network such as Japanese character handling, name handling for distributed resources, and communication technologies.

The history of JUNET started when University of Tokyo, Tokyo Institute of Technology, and Keio University started exchanging mails using UUCP on dial-up lines in October 1984. JUNET[10][13] connects local area networks at research institutes in Japan and it also provides users with the means of the worldwide communications via various international links.

There are some historical reasons which have prevented Japanese research and development communities from establishing a network for them. One of such a reason is that there have been not so many good computers for communication in the communities. Another example of such a reason may be that there used to be more restrictions in using the telephone lines than there are now. Since 1983, UNIX[3] has become popular in Japanese research and development communities, and this has encouraged the communities to start a simple network such as the one using UUCP[11] protocol over dial-up telephone lines. Applications used that time were electronic mails and electronic news.

At the time the three universities started the network, researches specially required in Japanese environment, such as Japanese language interface for communication applications, Japanese character code standard, and interconnection to existing research and development networks, were started to take place. The domain addressing over UUCP network was introduced in May 1985 with a system to gen-

[3] UNIX is a trademark of AT&T Bell Laboratories.

erate the address conversion software. High speed dial-up modems have been studied in order to increase the transmission rate of the communication. To achieve this purpose, UUCP enhancement with kernel driver development were done, and the efficiency using a dial-up line increased up to more than 13kbps. This encouraged us to migrate to TCP/IP protocol suite[3][4] even on dial-up lines as well as leased lines. As the result of the development, the dial-up IP link is providing as high as 7.5kbps in end-to-end transmission.

As for the internetworking between JUNET and other academic networks, two gateways are operated to exchanging electronic mails and news. One is Kokusai Denshin Denwa Co., Ltd. (KDD), an international telephone and telegraph company which serves a gateway between UUCP/USENET and JUNET, and University of Tokyo is providing a gateway function between CSNET[2] and JUNET.

One of the primary characteristics of JUNET is handling of Japanese characters in text messages. In order to provide the functions, a JUNET standard Kanji code was chosen and conversions between the network standard Kanji code and operating system Kanji codes are provided by the network application available in JUNET. The general computing environment for Japanese character handling are established as well in order to cooperate with the network environment. The statistics introduced in this paper shows drastic influences of Japanese character handling in computer network environment in Japan.

2 Current Status of JUNET

JUNET currently connects more than 250 hosts in 87 organizations. The geographic areas covered by the network expands from Hokkaido, the northern island, to Kyuushuu, the southern island, however, concentrations in Tokyo and Osaka areas are obvious as shown in Figure 1.

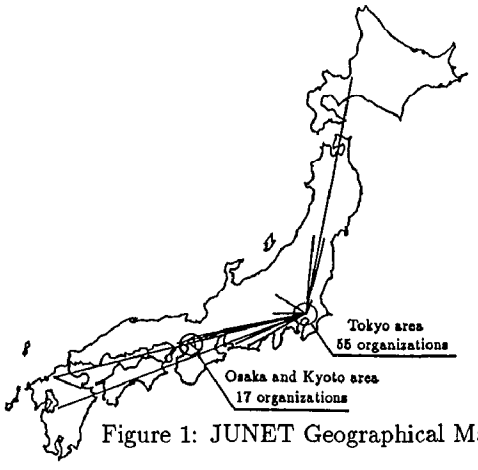

Figure 1: JUNET Geographical Map

Most of the links have been dial-up lines using 1200bps or 2400bps modems, although special mechanisms have developed for UNIX operating system and its communication software UUCP in order to use high speed dial-up modems with 9600bps or higher transmission rate. These mechanisms are also effective for TCP/IP protocols over dial-up telephone lines.

Organizations connecting to the network are universities, research laboratories of computer software/hardware companies, and research laboratories of telephone companies. All the functions are administrated by administrators at each of the institutes in totally volunteer basis.

2.1 Size of JUNET

Size of JUNET can be examined by various statistic values. The constant growth in number of organizations on the network is shown in Figure 2. One or two new organizations are being connected to JUNET every week.

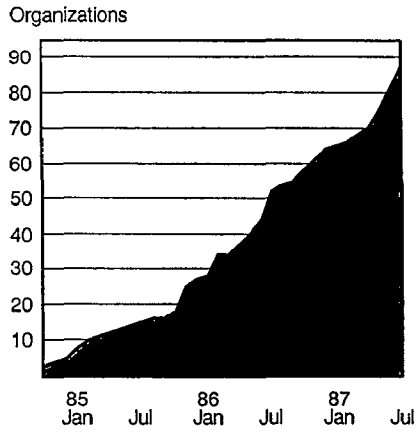

Figure 2: Number of domains

News articles are posted to the network constantly in the `fj` newsgroups which are currently distributed only within Japan. The number of articles posted has been growing as in Figure 3, and 1750 articles are posted in June 1987, as the latest example. Note that only about 10 percent of the articles are written in English alphabet including articles in ROMAJI, an alphabetical representation of Japanse language.

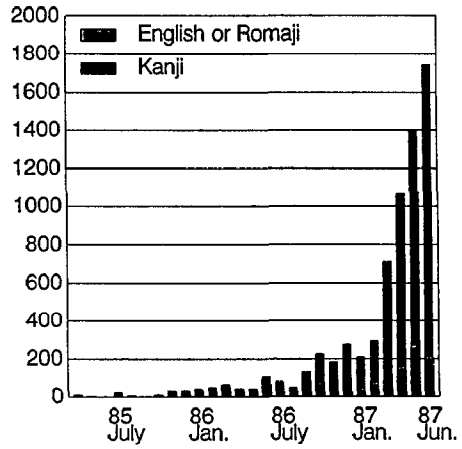

Figure 3: Number of Articles in `fj` newsgroups

Each of the JUNET gateways are handling 22 M bytes of articles in a month; 4 M bytes of `fj` newsgroups and 18 M bytes of USENET newsgroups.

One of the systems in the JUNET backbone handles about 200 M bytes of information in a month, and 85 percent of them are for the network news and 15 percent are for electronic mails. Since the network news are compressed to half of their size before the actual transmission, about 370 M bytes of text information is handled in one of the most busy systems in JUNET.

The international information exchanges are served by two gateways of JUNET; `ccut.cc.u-tokyo.junet` of University of Tokyo and `kddlab.kddlabs.junet` of KDD laboratories. The `ccut.cc.u-tokyo.junet` is on CSNET as `utokyo-relay` and is operated as the JUNET–CSNET gateway, on the other hand, the `kddlab.kddlabs.junet` is widely known in the UUCP network and is operated as the JUNET–UUCP network gateway.

The amount of international messages can be estimated by the total traffic at these two gateways. The example traffic of June 1987 are shown in Table 1.

Table 1: International Message Traffics

gateway	mail	news
`kddlabs.junet`	10MB	18MB
`u-tokyo.junet`	13MB	0
total	23MB	18MB

In summary, JUNET currently has approximately 41 M bytes of international traffic in a month, and it has been increasing about 3 M bytes per month in recent six months.

3 JUNET Domain Addressing

In the hierarchy of JUNET domain structure, a domain called '`junet`' is the top domain, although we are now preparing to employ ISO's country code (ISO 3116)[7] for Japan '`jp`' as the top domain name[12]. The second level domains are called sub-domains, and each of them represents a name of an institute or an organization. Lower level domains than the sub-domains may be determined at each of the sub-domains. In any cases, the lowest level domains are the names of hosts. The names of sub-domains usually are names well known to the society, but such names sometimes differ in intra/inter-national environment. Therefore, one or more names can be registered as synonyms for a sub-domain name to help users to address with general knowledge on the name of organizations. A name of a resource, thus, is defined in one of the domains.

There are one of the distributed name server in each of the domains which handles definitions and deletions of names using a database dedicated to that domain. A name server of a domain thus has a database to define names of lower level domains adjacent to the domain, or names of resources, such as names of mailboxes. The information held by each of the distributed name server is used in retrieving information of resource names and in accessing resources. By this concept, a resource can be defined in a logical domain; a mailbox can be defined even in the top domain, `junet`. This provides a name space which is well-matched to the naming concept of the real world yet providing consistency and efficiency of operations.

3.1 Design of JUNET message delivery system

There is one name server existing for a domain where a distributed resource name can be defined. In every host, at least one name server exits; a name server for a domain representing that host. Other than the one, name servers which represent domains located along a path from the root to this system in the domain tree can be existing in this system. Thus, name servers for the logical domains are managed by entities which are executed in a distributed manner.

A message delivery system using the above concept is implemented using sendmail[1] whose rule is generated by a rule generating system which plays a role of name servers of the JUNET naming concept.

Since the production rules of the sendmail system is differed site by site, the rules have to be generated at each site. To keep the consistency in the rules over JUNET sites, a configuration system to generate the necessary rules is designed and implemented. The configuration system to construct a name server receives information about domain names and about connections for communication among the name server. Then it generates sendmail rule as its output.

Information supplied to the configuration system are as follows:

Domain Names: List of domain names to be configured and names of domains defined in these domains.

Routing Information: Information about actual connections and name of mailers.

Mailer Definitions: Attributes of a mailer such as address formats and calling systax which the mailer is expecting.

The structure of the JUNET addressing system is illustrated in Figure 4.

In the input data, the following information about the system is described in the entries shown below:

name: A full name of this domain and its synonyms if any.

lower_level_name: A list of domain names defined in this domain.

neighbor: mailer (domain [cost])... *Neighbor* specifies a name of a system with which a direct connection is possible. A mailer used to transmit messages to the system is specified by *mailer*. A list of domain names defined in this domain are in *domain*. The optional *cost* field is used for each of the domains to represent a cost for sending a message to the domain. This is used to determine one route out of several possible routes.

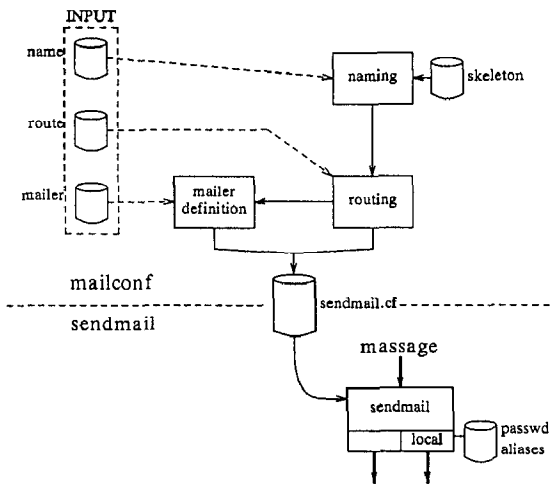

Figure 4: Structure of the Configuration System

The domain database contains relations between a physical link to a system and domain names which should be solved at that system. Other than the sendmail, the *rmail* command which receives messages through UUCP links was modified to handle JUNET addresses efficiently.

4 Japanese Character Handling

There always exist strong demands for Japanese character handling for computer software in Japan. Communication software can never be an exception: The statistics in Figure 3 shows most of the news articles are written in Japanese characters, or in *Kanji* codes. It is observed that the large amount of electronic mails are also exchanged using Kanji codes. Therefore, One of the remarkable characteristics of JUNET among academic networks in the worlds is Kanji message handling for text message exchanges. In order to discuss about this topic, some basic ideas about Japanese character handling have to be discussed beforehand.

4.1 Kanji code

The history of Kanji code handling in computers is rather short, and there still are several confusions about Kanji code itself. That is to say, there are some 'Kanji code standards' actually used in computer world. However, all the 'standards' refer to a single standard defined by JIS (Japanese Industrial Standard) X 0208 in their way of defining set of Kanji characters. JIS is always referred to because it can be introduced from any codes defined by International Standard Organization using ISO 2022 extension guidelines. It defines a Kanji code by two 7-bit bytes without using the most significant bits (MSBs). And there are several different Kanji codes exist for computer software because the switching method defined by ISO 2022 is not practical for random access to text messages.

The JIS X 0208 defines not only Kanji characters, but also English alphabets, digits, two types of 50 Kana characters (phonetic representation of Japanese language), and special characters. Among them, Kanji characters defined by JIS X 0208 are divided into two separated groups; one is called level one and another is called level two. The level one includes about 3500 Kanji characters and this set provides sufficient number of Kanji characters for most of ordinary texts such as technical writings. There although need more complicated Kanji characters for the advanced text applications such that text including names of persons, names of places, or text of literatures. For the purposes like these, JIS defines another set of Kanji characters called level two. In level two, there are about another 3400 characters.

In total, about 7 thousand Kanji characters are necessary for providing Japanese character capability in computer. Obviously, this requires more than one byte to represent, and representing a single character with two bytes is a standard concept. The important issue here is that we still need to use ASCII codes in computers as well as Kanji codes. This means we have to establish a way to handle a mixture of ASCII codes and Kanji codes. In order to solve the problem, there have to exists a way to distinguish ASCII codes sequences from Kanji code sequences.

In order to distinguish the Kanji sequences from sequences of ASCII representation of English alphabets and special characters, there are three major methods existing:

1. JIS X 0208 codes with JIS X 0202 (ISO 2022)
2. Extended Unix Code (EUC)
3. Microsoft Kanji Code (Shift-JIS)

By 1., a Kanji sequence is surrounded by a introducing escape sequence of Kanji characters (`ESC-$-@` or `ESC-$-B`) and introducing sequence of English alphabets (`ESC-(-J` or `ESC-(-B`).

2. is originally defined by AT&T UNIX Pacific to provide internationalized version of UNIX operating system[9] and is becoming to be a standard way to represent Kanji codes in UNIX operating system in Japan. In the EUC, the way to distinguish Kanji code sequences is somewhat different from the one of 1. By assumption that there never exists an ASCII code with its most significant bit on, a Kanji character can be distinguished by its significant bit; setting it on for every byte representing a half of a Kanji code.

3. was originally defined for CP/M on personal computers by a Japanese subsidiary of Microsoft Corporation. This is now a defacto standard for personal computers and is also supported in some of the Japanized UNIX environments. In the Microsoft Kanji codes, Kanji characters are mapped by some function so that first byte is in ranges of `0x81` – `0x9f` and `0xe0` – `0xff`.

In order to cooperate with the ISO standard method to handle character codes, and to utilize existing software which sometimes strips the MSBs off, we decided to use JIS X 0208 two 7 bit codes surrounded with escape sequences as the network standard. This decision requires conversion functions from JIS X 0208 to local operating system Kanji codes, namely EUC/JAPAN and Microsoft Kanji code.

4.1.1 Kanji code handling

In order to design network applications using Kanji codes, The following discussion have to be made regarding the average environments of existing JUNET systems.

1. There are several kinds of character codes including above three codes which are actually used in operating systems as internal codes to represent Japanese characters including Kanji characters. Among them, JIS X 0208 is not practical for internal code because of the complicated operations of switching modes when seeking a character in a byte sequence; random access to a sequence of byte is impossible. Thus, two 8 bit codes without escape sequence is preferable to two 7-bit codes with escape sequences as internal code for an operating system.
2. Since we have decided to use JIS X 0208 as the JUNET network standard Kanji code, we have to provide conversion mechanisms from JIS X 0208 to internal code of operating systems such as EUC and to Mircosoft Kanji code.

4.1.2 JUNET approaches

By assuming the above issues, we have developed the following environment for JUNET.

Kanji code: As we have described before, we are using JIS X 0208 with ISO 2022 / JIS X 0202 extension guide line as a network standard Kanji code. Escape sequence introducing JIS X 0208 can either be `ESC-$-@` or `ESC-$-B`. These are two definitions of two minor versions of JIS X 0208 and both are legal in a sense of standard definition. Escape sequence introducing ASCII code is `ESC-(-B` and introducing ROMAN character code of JIS is `ESC-(-J`. There are not much differences between ASCII and ROMAN character code, except that image representations for a few characters are different. We therefore treat both of them equally. Note that the default code set for JUNET text message is ASCII code; there is no introducing escape sequence necessary if a text starts with a ASCII code.

Control characters: According to the ISO 2022 / JIS X 0202, any control characters appear in both Kanji sequence and ASCII sequence, however, deep backtracking may be necessary to determine the character mode when a file is accessed randomly. So we decided not to allow control characters appeared within Kanji code sequence; they can only appear in ASCII code sequence. This rule contributes to make software which handles text messages to be simple and transparent in terms of internationalizing because a function to find a control character can transparently be defined.

Single byte Kana code: There are another code set for representing Japanse character in a single byte; JIS X 0201. In this code, only Kana characters are represented. This code set used to be used in mainframes because Japanizing of software was easy. Since we now have Kanji code set anyway and all the Kana representations are included in JIS X 0208 using 2 bytes, we eliminate usage of JIS X 0201 to avoid the complexity caused from handling of three different code sets at a time.

Network software: In order to provide an environment for Japanese text message capabilities in JUNET, we have modified the following software to adopt the above strategies:

- Bnews[5] was modified to pass the escape sequences which used to be stripped off in the original version. Conversion functions from JIS X 0208 to major internal Kanji codes are also added. Other network news interfaces such as rn and vn have been modified, too.
- MH[14] has been added the code conversion facility between the network standard code and the local code.
- GNU emacs and MICRO emacs were modified to handle Kanji codes so that we can edit Japanese mails and articles.
- X-window also was modified to represent Kanji characters. Level one Kanji fonts have been *handmade* and posted to JUNET[16] so that a user can read and write Kanji characters without a special Kanji terminals.

An example to picture example JUNET environment of Japanese character handling is shown in Figure 5.

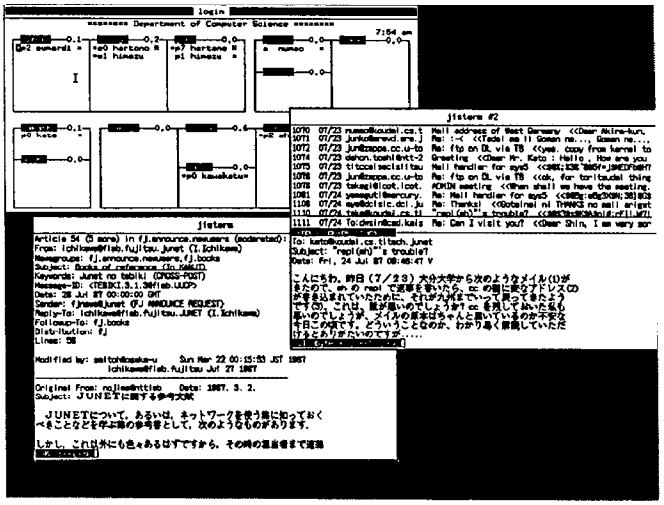

Figure 5: Example of JUNET Kanji Environment

5 Communication Software Technologies

The primary goal of the development of JUNET as the first step was to construct a network providing functions for

text message exchange over research communities in Japan. Since JUNET project is a volunteer project to provide basis of researches on network communication, UUCP protocol over dial-up link was our choice to start the network because of the popular, easy to set-up and inexpensive features of UUCP technology. The dial-up modems used have been upgraded from V.21 300bps to V.22bis 2400bps and we have almost succeeded to eliminate the V.21 and V.22 1200bps modems.

However, as the size of the network has been grown and traffic has been increased as described in chapter 2. 2400bps communication is not practical at most of the major systems on the network, especially with limited resources of telephone lines at universities. In order to solve the problem, three major steps have been planned for the network; i) use of high speed dial-up modems, ii) dial-up IP links over the high speed dial-up modems, and iii) use of high-speed leased lines or X.25 packet switching network.

The purpose of the first step is to develop the most efficient usage of the high speed dial-up modems for UUCP usage; including protocol modification and kernel modification of UNIX operating system. This mechanism improves transmission efficiency at a system by 3~5 times compared with UUCP over V.22bis modems.

Utilizing technologies established by step i), major transport protocols such as TCP/IP over the dial-up link becomes practical. The step ii) thus designed and developed to provide better communication possibilities for network development as well as to provide better environment for users.

Step ii) above introduced us with technologies to construct an IP-based network; problems such as name serving, address managing and routing have to be studied and solved. This encouraged us to start working on developing a network based on permanent (virtual) circuits such as high speed leased line or X.25 based packet switching network.

5.1 UUCP modifications

The high speed dial-up modems in this paper means modems which have dial-up functions and realize error free and high speed data transmission by a modem-to-modem protocol. The actual transmission rate of these modems depends on size of the data transmitted because of the built-in protocol.

Since the host computers on JUNET varied in vendors and types of software and hardware including versions of UNIX operating system, the flow control mechanism in serial line communication are also varied. By experiences in JUNET, we found that we can hardly develop a general or portable communication mechanism by relying on flow control mechanism provided in vendor products especially for high-speed communication in serial ports.

Thus, we have employed DC1/DC3 flow control mechanism to achieve our purpose of efficient usage of the high-speed modems[15]. The basic strategies of the mechanism are; to develop a UUCP protocol which escapes DC1, DC3 and DLE ascii control characters, and to develop a *line discipline* module for UNIX tty driver in order to handle DC1/DC3 properly.

5.1.1 UUCP/j-protocol

There are some existing protocols for UUCP such as *g, f, x*, and *t*. The *j-protocol* developed for JUNET is similar to the *f* in purpose but has two primary characteristics: It allows 8 bit bytes and escapes DC1, DC3, and DLE ascii characters. Unlike *f-protocol* which was designed for communication using PADs, 7 bit encoding is not necessary for high-speed modem communications. DC1 and DC3 characters have to be escaped to make the link transparent as we use these characters for the flow control mechanism between the machine and the modem. The DLE was also escaped because this character is often treated specially in some serial communication path.

5.1.2 Utty — Line Discipline

UUCP normally sets UNIX tty driver to be in *raw* mode which provide transparent communication through serial ports. The *tandem* mode of tty driver exists where the driver suspend transmitting output characters by receiving a DC3 character. For the purpose of utilizing the high-speed modem with DC1/DC3 flow control, the driver have to generate a DC3 character when more than a certain amount of characters are in input queue to avoid overflow in the queue.

The line discipline developed for UUCP, *utty* (Uucp TTY), is a driver which has a 1024 bytes ring buffer for input data with a high water mark and a low water mark. When the input data in the buffer reaches the high water mark, the driver generate a DC3 onto the output queue, and a user process receives the data and the amount of the data in the buffer decreases less than the low water mark, it generates a DC1. Functions to control the output transmission is same as one of normal tty driver's.

5.1.3 Discussions

The UUCP communication using the *j-protocol* over *utty* line discipline has been working in JUNET for several months. The high-speed modems currently used are TrailBlazer of TELEBIT corporation and AX/9624c of Microcom corporation. The actual performance of the mechanisms for various data size of UUCP is shown in Figure 6.

This also shows the result of V.22bis 2400bps modem and V.22 1200bps modem. The statistics were collected from daily operations of a major JUNET system.

The maximum data transmission rate for the high-speed modems is 13kbps for TrailBlazer and 12kbps for AX/9624c. These performance could be achieved only for a large data transmission; the drastic increasing of the transmission rate for larger data is caused from built-in protocol

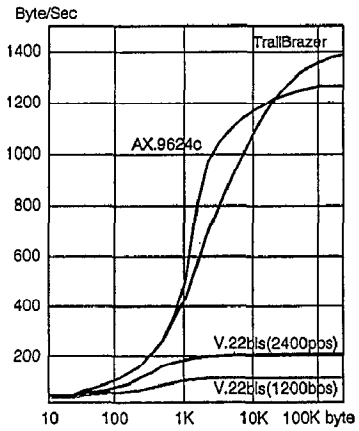

Figure 6: Performance using j-protocol UUCP/utty

mechanism. The port utilization time decreased to 25 percent of one by V.22bis modem. This is because most of the JUNET traffic is for transmission of news articles as described in chapter 2, and the news articles are spooled and concatenated (batched) into a large data which increases performance of the communication mechanism introduced in this chapter.

5.2 Dial-up IP Link

As the result of the high-speed dial-up modems described above, we could expect transmission rate of more than 10kbps on a dial-up link. This encouraged us of using a dial-up link for various transport protocols such as TCP/IP. IP connections over a dial-up link have been usually considered for personal computers at home[6] however, they are still effective for easy construction of an internetworking among universities, especially on the achieved performance with the high-speed modems.

The system for dial-up IP link thus was designed and developed[8]. The first version of the system was designed using the *utty* line discipline described in the previous section. And with the experiences from the first version, the second version was designed and developed which achieves improved performance by eliminating data movements between the user and the kernel space of UNIX operating system. Both of the systems are developed on SUN OS version 3.2 which is compatible with 4.2BSD UNIX.

5.2.1 The First Version

The first version of the system consists of 4 major modules and the *utty*. The *dldriver* resides in the kernel and receives and sends IP datagrams with the IP module which also exists in the kernel space. It has communication entities as a "special file" in the UNIX file space called /dev/dl*x*. The major portion of the system, *dldaemon*, is a permanent user process which reads/writes information from/to the *dldriver* and allocates *utty* driver. The character mappings to escape DC1/DC3 and other control characters are also performed by the *dldaemon*. Another module called *dlattach*, which is a user command, is invoked at login time to pass the information about the serial port to the *dldaemon*. Figure 7 shows the structure of the system.

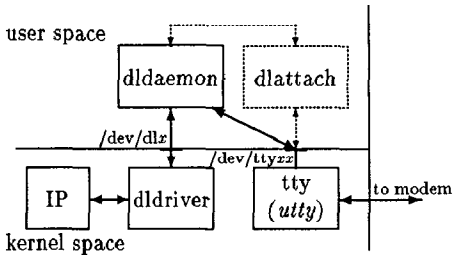

Figure 7: Structure of the Version one

A connection established with the system is a point-to-point connection between a master system which initiates the connection and a slave system. The following example of sequence shows an overview of the procedure and the functions of the system:

1. The *dldaemon* which is invoked at a boot time resides on the system permanently until the operating system goes down. As soon as the process is initiated, it registers a network interface (*if*) called dl*x* and sets up the routing information in the kernel space so that the IP module can pass IP fragments with addresses routed toward the link to the *dldriver*. This initiating procedure have to be done at both sides of the systems.
2. When an IP fragment is selected to the dl interface by the routing function in the IP module, the *dldriver* passes the fragment to the *dldaemon* running in the user space via a special device.
3. If no connection for the dial-up link is established, the *dldaemon* starts dialing to make a connection to the remote site. The dialing is done by a part of the public domain dial program.
4. According to the scenario description for dialing, the master *dldaemon* logs-in to the slave system and invokes the *dlattach* command.
5. The *dlattach* provides the running *dldaemon* on the slave system with the information about the serial line port and about the caller's system via Unix domain socket, as shown in the dotted line of Figure 7.
6. When the physical connection is established, each of the *dldaemons* at the both systems generates a system call to switch a line discipline to the *utty*. Finally, they complete establishment of an dial-up link ready for IP communication between the systems.

The resulted performance of the first version with two TrailBlazers was approximately 3kbps for the end-to-end communication by ftp and the utilization of the line was fairly low. There are two reasons which decrease efficiency: The

dldaemon waits for input both from *dldriver* and *utty* using the *select* system call and this is a bottle neck of the communication. Another reason is that the data movement between the kernel space and the user space obviously causes overheads.

5.2.2 The Second Version

The second version of the system eliminates the data movement between the kernel space and the user space by defining protocol interface between the *dldaemon* and the *dldriver*. The *dldriver* now includes a line discipline very similar to *utty*, but IP fragments are passed between the tty driver and the *dldaemon* directly using a character buffer called *clist*. The communication between the IP module and the *dldriver* is done via *mbuf*, a buffer control mechanism for network modules in the kernel. This also eliminates overhead of data movement. The structure of the second version of the system is illustrated in Figure 8.

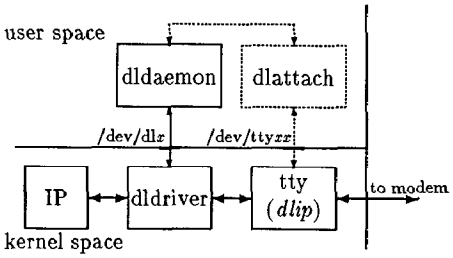

Figure 8: Structure of the Second Version

A connection is terminated by *dldaemon* when one of the following conditions is detected; i) 60 seconds passed without any IP fragment transmissions, or ii) missing of 'I am alive' control character from the remote system for 24 seconds. This character is generated every 8 seconds to indicate that the local *dldaemon* is active to the remote *dldaemon*.

Interactions between the *dldriver* and the *dldaemon* are performed by exchanging messages which defines protocol interface of *dldriver*. The actual data are no longer passed to *dldaemon*, however, some messages generated by *dldriver* are sent to *dldaemon* as follows:

DG_REC indicates arrival of an IP fragment from the remote site.

DG_SEND indicates that an IP fragment passed to the serial line.

IAA_REC indicates that the *dldriver* receives a control character of 'I am alive' from the remote site.

DIAL_REQ requires dialing to the remote site.

On the other hand, commands generated by the *dldaemon* for the *dldriver* are as follows:

OPEN_CONNECT indicates the establishment of physical connection by completion of the dialing.

SEND_IAA generates 'I am alive' control character to the remote site.

CLOSE_CONNECT forces termination of the connection and re-initialize the modules.

DIAL_FAIL indicates the dialing procedure fails and forces the queued fragments discarded as well as the re-initialization of the modules.

5.2.3 Discussions

The measured performance of the system with two TrailBlazers is about 7.5kbps for end-to-end data transmission by ftp, which achieves most efficient transmission rate for serial ports. Time consumed by the dialing sequence may be critical in terms of a timeout interval of upper layer protocol. By our experience, however, the dialing time has never exceeded the TCP's connection time out yet.

There is no dynamic routing control mechanism employed for this system. It is necessary for this system to develop intelligent routing functions with considerations of speed and availability of links.

6 International Information Exchange

As described in this paper, number of messages exchanged in JUNET has been increased rapidly, and number of messages exchanged internationally via the two gateways have also been increased as well. Users of other networks, however, sometimes complains about the insufficient information on world networks generated form JUNET. The primary reason for the complains is obviously caused by the preference of Japanese characters among JUNET users.

The development of general purpose software to handle Japanese characters in non-special hardware environment as well as development of the hand-made Kanji fonts encourages us to distribute JUNET Kanji messages to other countries. The experimental delivery of the domestic newsgroups to abroad has started in 1986 for some universities.

On the other hand, submission of news articles from USENET environment to JUNET can be achieved by adding a newsgroup called `fj.misc` to the newsgroup list of an article. This is efficient because all of the JUNET sites are subscribing `fj` newsgroups while some sites are not subscribing USENET newsgroups. The news article posted this way is handled as a JUNET news article inside Japan.

It is known that the addresses of JUNET are properly handled at both `relay.cs.net` and `seismo.css.gov`. Therefore,

- *user%domain*`.junet@relay.cs.net`
- *user%domain*`.junet@seismo.css.gov`

are the most popular style of addresses from the ARPA Internet name space. However, non university users in Japan

can not use the link between `ccut.cc.u-tokyo.junet` and `relay.cs.net`. So mails to non university people should be routed to `kddlab.kddlabs.junet`. In order to contact JUNET administrators, `junet-admin@junet` is the address.

7 Conclusion

Development of JUNET started in October 1984 and various researches on computer networking and distributed environment have been actively done with the rapid growth in the size of the network. Among them, name management functions to construct a hierarchical domain name space, Japanese character handling, and communication technologies using the high-speed modems have been focused as primary research concerns.

The actual work for the addressing and routing of JUNET text messages is achieved by a name server concept and its implementation. This system receives control messages to specify the information about logical domain name, connections and methods to use to deliver messages, and generates a sendmail rule set. This software provides an environment where the logical naming definitions and physical routing issues are clearly separated so that reliability, efficiency, extensibility, and flexibility of communication in the network are simultaneously achieved.

Internationalization of computer software is one of the most important issues in computer science, and some works have been achieved in JUNET communication software. In JUNET, electronic mail and news software are designed to enable 16bit Japanese character handling, based on the studies and discussions on general computing environment using Japanese text. The clear separation of network Kanji code and internal operating system code employed in JUNET software provides a transparent environment on Japanese character handling in computer networks. Availability of Japanese messages obviously encourages users to exchanges messages over the network.

Requirements for higher transmission rate of the dial-up lines, which was chosen to start the network, have arisen and the use of the high-speed modems have been studied. A new UUCP protocol and tty driver enhancement are developed for the purpose. As the result, more than 13kbps UUCP data transmission rate is achieved. This encouraged us to migrate onto TCP/IP suite using dial-up lines as well as using leased lines. Performance of the implementation of the dial-up IP link is about 7.5kbps which is practical enough to construct a distributed environment over a widely interconnected network environment.

Progresses of JUNET technologies discussed in this paper lead us to many topics of future studies such as:

- Enhancement of name servers which handles general distributed resources.
- Establishment of gateway technologies such as optimal routing strategies based upon constructions of IP-based network using the dial-up IP link and IP over leased lines.
- Supports of multi-media message exchanges enhances the environment with multi-language supports currently achieved.

The technologies developed in JUNET can generally be used to construct a network interconnection with inexpensive cost.

Acknowledgment

The authors would like to express our gratitude to many researchers who have been contributing to establish the JUNET environment described in this paper. Among them, special appreciations go to Youichi Shinoda, Keisuke Tanaka, and Hiroshi Tachibana for their efforts in developing JUNET software. The dial-up IP link was achieved as a result of discussions with members of the JUNET project especially with Susumu Sano. Professor Haruhisa Ishida and Professor Yoshihiro Tohma for their valuable advices and encouragements. We would like to thank these people on behalf of the project.

References

[1] ALLMAN, E. Sendmail – An Interconnecting Mail Rerouter, Version 4.2. In *UNIX Programmer's Manual, 4.2 Berkeley Software Distribution*, Univ. of Californa, Berkeley, 1983.

[2] COMER, D. The computer science research network CSNET: A history and status report. *CACM 26*, 10 (October 1983).

[3] (ED.), J. P. *Internet Protocol – DARPA Internet Program Protocol Specification.* Tech. Rep. RFC 791, USC/Information Sciences Institute, 1980.

[4] (ED.), J. P. *Transmission Control Protocol – DARPA Internet Program Protocol Specification.* Tech. Rep. RFC 793, USC/Information Sciences Institute, 1981.

[5] EMERSON, S. L. USENET: A bulletin board for UNIX users. *BYTE* (September 1983).

[6] FARBER, D. J., DELP, G. S., AND CONTE, T. M. *A Thinwire Protocol for connecting personal computers to the INETERNET.* Tech. Rep. RFC-914, Univ. of Delaware, 1984.

[7] ISO. *Codes for the Representation of Names of Countries.* Tech. Rep. ISO-3116, International Standard Organization, 1981.

[8] KATO, A. Network news system in JUNET. In *Proceedings of the 35th Annual Convention* (1987), IPSJ. (In Japanese).

[9] KOGURE, H., AND MCGOWAN, R. A UNIX System V STRAMS TTY Implimentation for Multiple Language

Processing. In *USENIX Summer Conference Proceedings* (1987), USENIX.

[10] MURAI, J., AND ASAMI, T. A network for research and developement communications in Japan — JUNET —. In *Proceedings of First Pacific Computer Communications Symposium* (1985).

[11] NOWITZ, D. A., AND LESK, M. E. *A Dial-Up Network of UNIX Systems.* Tech. Rep., Bell Telephone Laboratories, August 1978.

[12] POSTEL, J. *Domain Requirements.* Tech. Rep. RFC 920, Univ. of Southern California, 1984.

[13] QUATERMAN, J. S., AND HOSKINS, J. C. Notable Computer Networks. *CACM 29*, 10 (October 1986).

[14] S. BORDEN, R. S. G., AND SHAPIRO, N. Z. *The MH Message Handling System: User's Manual.* Rand Corporation, 1979.

[15] SHINODA, Y. Dialup network by high speed modems in JUNET. In *Proceedings of the 35th Annual Convention* (1987), IPSJ. (In Japanese).

[16] TACHIBANA, H. PD kanji font (tools and ascii fonts). Network news posted to JUNET newsgroup `fj.sources` as `<1676@rika.cs.titech.JUNET>`, July 1987.