An alternative method for connection between TANet and the campus network for High and Junior High & Preliminary School — A solution to the lack of the IP subnet problem of TANet

陳鴻彬　　趙涵捷　　林信鋒

國立東華大學電子計算機中心
花蓮縣壽豐鄉志學村大學路二段一號
E-mail: hbc@cc.ndhu.edu.tw

Abstract

Owing to more and more powerful applications developed on public domain, people perceive that they can do many things on Internet, which causes the Internet rapidly growing during these years. The problem of exhaustion of IP addresses has thus been awaken in the entire Internet community. Many papers had been proposed for solving this problem [1][2][3], and stricter guidelines for the IP subnets assignments had been made [4][5]. Therefore, it is no longer easy to get a large number of IP subnets for TANet.

In this paper, we will propose a workable scheme that can provide almost every kinds of services on TANet from a small network, like High and Junior High & Preliminary school network, which only consumes one legal IP address to connect to TANet. This scheme uses a Unix host as a fire-wall to connect between TANet and the small campus network, each client PC behind the fire-wall can use the E-mail, Gopher, Archie, Ftp, BBS, News, WWW... services via the fire-wall without consuming a legal IP address. High and Junior High & Preliminary school's small campus networks can then rapidly connect to TANet, and no longer need to wait for the IP subnets to be assigned from TANet administrator. This can solve the lack of IP subnets problem of TANet.

1. Introduction

Observing the current TANet connection situation, we know that the leased-line connection between routers is the major connection method on TANet add citation. In detail, there are many regional network centers scattered around the island, each regional center is connected with a router serial port via a high speed leased-line (64K, 256K, T1, or higher), these form the TANet backbone. Each non-regional center site uses a router serial port to connect to a router serial port of the local regional center via a lower or equal speed leased-line than backbone. For each connected site, some registered IP subnets should be assigned to this site from TANet. It causes some IP addresses to be wasted, because a class C IP subnet can support 256 IP addresses, but most of the class C subnets on TANet have less than 200 nodes. In this paper, we will discuss another method for connection between a small campus network and regional center only consumes one IP address of the local regional center, and each pc client in this small network can use most of the services provided by TANet without consuming a legal IP address of TANet.

1.1 Scheme Development Background

The National Dong-Hwa University (NDHU) was established on July 1994, there was no campus network established at that time, and the leased-line connection between NDHU and NHLTC (National Hua-Lien Teacher's College) had not been installed yet. In order to connect to TANet, besides applying for a 64K leased-line for connecting between NDHU and NHLTC, we also requested for some IP subnets, but these two requests need time to be process. To provide E-mail service and other TANet services to users as soon as possible on campus, we built a temporary network and used the scheme described through this paper to connect to TANet promptly. We had used this alternative method to send E-mail and other TANet services (Archie, Ftp, Gopher, BBS, News, WWW,...) for several months before the leased-line connection was established.

1.2 Scheme Modeling

As the Fig.1 shown, suppose a small campus network constructed by some PCs and one or

more Novell PC rooms on the network. Now, we want to connect this small network to TANet, and hope each user can access almost every kinds of services provided by TANet from any PC on this network.

The classical TANet connection method is to use a leased-line to connect between local router and the remote router of regional center, as the Fig.2 shown. For each IP subnet domain, this method needs to apply an IP subnet for it from TANet that may cause some legal IP addresses wasted. It also needs to occupy one router serial port of regional center. Another disadvantage of this method is to wait for the leased-line to be established and IP subnets to be assigned. Thus it will delay the schedule for all High and Junior High & Preliminary schools connecting to TANet.

The previous connection method described in this section need one or more IP subnets to be assigned. However it is more difficult to get enough IP subnets for all High and Junior High & Preliminary school campus networks than before. This is an unfavorable factor to introduce TANet to each High and Junior High & Preliminary school. For solving this lack of IP subnets problem, we propose an alternative scheme to connect these small campus network to TANet. For the model of a small campus network shown in Fig.1, Fig.3 illustrates how this method work on the model. We use a Unix host to connect to a terminal server port of the local regional center through a dial-up telephone circuit rather than a leased-line. And we run a TCP/IP over serial line protocol on both the local Unix host and the remote connected terminal server port of the local regional center. Then the Unix host can get a preassigned legal IP address from the remote connected terminal server port. The Unix host can now be considered as a fire-wall machine for the small campus network. The small campus network which connected to the fire-wall machine can use any of unregistered IP subnet, and each PC client of the network can access TANet via the fire-wall machine. We will illustrate this scheme step by step under this model through out the following sections. First, we will discuss that how to configure the Unix host to be a fire-wall for the small campus network. Secondary, we will discuss how each PC can use the variety of TANet services via the fire-wall.

2. Configure the Unix host as a Fire-wall

The Unix host use in our model can be a Unix workstation or a powerful PC (like Pentium PC) which runs a Unix operating system (e.g. Linux) on it. Here we will use a SUN workstation to run SunOS 4.1.x. to explain our method.

2.1 Connect to TANet via a dial-up telephone circuit

To configure the Unix host as a fire-wall for the small campus network, we should connect the Unix to TANet at first. In our propose scheme, we connect the Unix host to a port of the terminal server of the local regional center via a dial-up telephone line. We should choose a communication software to provide the dial-up function. There are many communication softwares like C-Kermit, X-modem, Y-modem and Z-modem can provide the dial-up function. Here we choose the C-Kermit to run on SunOS 4.1.x as an example for demonstration. This software can be obtained from ftp://lieca.ccu.edu.tw/pub1/unix/utility/ckermit.tar.gz. The steps to install this software are as the following [6][7]:

1. Uncompress and unpack the software:
   # gzip -d ckermit.tar.gz ↵
   # tar xvf ckermit.tar
2. Compile the software for SunOS 4.1.x:
   # make sunos41 ↵
3. Move the binary execution file wermit to the /usr/local/bin, and named kermit:
   # mv wermit /usr/local/bin/kermit ↵

The SUN workstation has two serial ports, say A and B. Suppose we connect the modem to the serial port A. After the above steps has been finished, the workstation is ready to use the dial-up function. For automatically dialing-up, we edit a script file, say /.kermrc, we should specify which modem type is chosen, which serial port the modem is connected, what baud rate is selected, the dial-up phone number, and all other necessary information. For more information about kermit, please type "man kermit". Here is a workable example of a script file that we used before :

```
set modem hayes
set line /dev/ttya
set speed 9600
dial 234162
connect
```

Now, we can type the command "kermit" on the SUN workstation to automatically dial-up to a port of the terminal server of local regional center.

2.2 Run a TCP/IP over serial line protocol

In the last session, we have discussed how to connect a Unix host to TANet via a dial-up telephone circuit, but this method can support only one user to use one telnet session to access remote machines on TANet via the remote terminal server at one time. It is obviously not our goal. The goal we want to obtain is that more than one user can access the TANet services from any PC located in the small campus network via the Unix host simultaneously. Thus, we should configure this connected dial-up link to simulate a leased-line connection link. Running a TCP/IP over serial line protocol is a solution to meet our requirements.

TCP/IP can run over a wide variety of physical media. The media can be Ethernet cables (as in local Ethernet), or telephone circuits (as in a wide area network). In this section, we will describe how to configure a network interface to use a telephone line, it needs to run either one of the two serial line protocols: Serial Line IP (SLIP) [8][10] or Point-to-Point Protocol (PPP) [9][10].

2.2.1 Serial Line IP Protocol — SLIP

SLIP is a simple protocol that allows isolated hosts to link via TCP/IP over the telephone network. It defines a simple framing scheme to send a datagram across the serial line as a series of bytes, and it uses special characters to mark such that the remote end can group a series of bytes as a datagram. Thus the IP datagrams can be transferred between the two ends via SLIP over a telephone circuit. SLIP is available for most Unix systems. For some of Unix systems, SLIP is as part of the operating system (like IBM AIX); or as part of the TCP/IP package (like SCO Unix). For those Unix systems that SLIP is not included, more effort to configure the SLIP software is required. We will detail the steps to install the SLIP software on SunOS 4.1.x, because that the SunOS 4.1.x is the most popular Unix system on TANet currently.

To install the SLIP software on SunOS 4.1.x, we need to get the source from ftp://nctuccca.edu.tw/UNIX/simtel/network/slipware.tar.gz via anonymous ftp. Then we need to use gzip to uncompress the tar-file and type " tar xvf slipware.tar " to unpack the software. Once the file is uncompressed and unpacked, it creates a directory called slipware with tar-files slip-4.0.tar and yapt5.5c.tar. Unpack these two tar-files which is similar to the way to unpack the slipware software, two sub directories slip-4.0 and yapt5.5c are created:

slip-4.0	Slip-4.0 is the version 4.0 of SLIP for SUN workstation running SunOS 4.x.
yapt5.5c	This is Yet Another Patch Tape (YAPT) 5.5c from Sun. It provides fixes for serial port problems.

The kernel of SunOS does not support SLIP. To make the SLIP available for SunOS, the kernel needs to be rebuilt. We separate the kernel rebuilding steps as three parts: Modifying System Files, Patching the Kernel, and Rebuilding the Kernel [10]. After rebuilding the kernel, the SLIP network interface should be configured for making a SLIP link [10].

Modifying System Files

For convenience, assume the slipware directory has been created under /usr/local. Then the directory, /usr/local/slipware/slip-4.0, contains the SLIP files. First, we should copy the required SLIP files from this directory to the appropriate sys directory. The steps are as followed:

```
# cd /usr/local/slipware/slip-4.0 ↵
# cp tty_slip.c /sys/os/tty_slip.c ↵
# cp slip.h  /usr/include/sys/slip.h ↵
# cp slip.h /sys/sys/slip.h ↵
```

Next, modify the two system files: /sys/conf.common/files.cmn and /sys/sun/str_conf.c such that they can be referred to our newly installed SLIP files. For /sys/conf.common/files.cmn, insert:

```
os/tty_slip.c          optional    slip
```

For /sys/sun/str_conf.c, inserts following three pieces of code:

```
---
#include "slip.h"
---
#if NSLIP > 0
extern struct streamtab slipinfo;
#endif
---
#if NSLIP > 0
    { "slip",          &slipinfo },
#endif
---
```

Patching the Kernel

Before rebuilding the kernel, we must apply the yapt5.5c patches to fix for serial problems. To install the patches, simply changes directory to the yapt5.5c and run install:

```
# cd /usr/local/slipware/yapt5.5c ↵
# ./install ↵
```

Rebuilding the kernel

Now that the SLIP system files and kernel patches are installed, we need to rebuild the kernel in order to run SLIP. For rebuilding the kernel, we should modify the kernel configuration file. We copy a kernel configuration file named GENERIC to a new configuration file named SLIPPERY that we are going to modify. Our example machine for illustration is SUN SPARC 10 (or 20) workstation:

```
# cd /sys/sun4m/conf ↵
# cp GENERIC SLIPPERY ↵
```

We edit the new configuration file SLIPPERY. Modifying the ident statement to identify the new configuration and add a pseudo-device for SLIP:

```
ident           "SLIPPERY"
---
pseudo-device           slip5
```

Now, we can rebuild the kernel using this new configuration file:

```
# /etc/config SLIPPERY ↵
# cd ../SLIPPERY ↵
# make ↵
```

Then, the new kernel will be compiled:

```
cc -sparc -c -o -Dsun4m -DSLIP ...
```

After this step is finished, the new kernel vmUnix is created in the current directory. Use this new kernel to replace the original kernel /vmUnix and reboot the system.

```
# mv /vmUnix /vmUnix.org ↵
# mv ./vmUnix /vmUnix ↵
# sync ↵
# reboot ↵
```

Configuring the SLIP Interface

SLIP is now installed in the kernel, but the SLIP network interface is not yet configured. The slipware software needs to use the sliplogin command to configure the SLIP interface. Before using it, we must compile and install it with set-uid permission. Following is the sample of installing sliplogin:

```
# cd /usr/local/slipware/slip-4.0 ↵
# make ↵
cc   -DLCKDIR=\"/var/spool/uucp/LCK\"
 target sun4 -o sliplogin sliplogin.c
# cp sliplogin /usr/local/bin ↵
# cd /usr/local/bin ↵
# chmod 4755 sliplogin ↵
```

We had connected to a terminal server port of the regional center via a dial-up telephone circuit in the session 2.1. For convenience, let's consider the network topology shown in Fig.4. Now, we want to configure the SLIP interface for SUN workstation such that it can work as a fire-wall for the small campus network.

To do this, we first need to run SLIP on the remote connected terminal port to get a preassigned legal IP address for the local fire-wall. For this example, assume the preassignment IP address is 192.192.6.106. Then, we need to use the sliplogin command to configure the SLIP interface for the fire-wall. The sliplogin command format likes this:

```
# sliplogin 192.192.6.110 192.192.6.106 <
/dev/ttya & ↵
```

Where the address 192.192.6.110 is the IP address of a Unix host on the regional center, as illustrated in Fig.4.

Now, we have made a SLIP link between the local fire-wall of the small campus network and the regional center via a dial-up telephone circuit. This SLIP link will work as a leased-line connection link. The remainder we need to do is to add the default route for the fire-wall, so that each fire-wall login user can access to other TANet hosts. For our example, the command for adding the default route likes this:

```
# route add default 192.192.6.110 1 ↵
```

If a network user of the small campus network wants to telnet or ftp to other TANet hosts, he can first telnet to the local fire-wall from his unregistered PC via local campus network and use the telnet or ftp command of the fire-wall to access other TANet hosts. Because that he can use telnet command on the fire-wall to access other TANet hosts, he can also use the BBS and Archie services. This is done by telnet to those servers on TANet. If he wants to use Gopher service or read News, he also can run gopher client or run rtin command on the fire-wall to reach his goal.

2.2.2 Point-to-Point Protocol — PPP

SLIP has been introduced at last session, but it has some disadvantages:

- The SLIP protocol does not define any link control information that could be used to dynamically control the characteristics of a connection. Therefore, SLIP systems must assume certain link characteristics. Because of this limitation, SLIP can only be used when both hosts know each other's address, and only when IP datagrams are being transmitted.
- SLIP does not compensate for noisy, low-speed telephone lines. The protocol does not provide error correction or data compression.

To address the two problems of SLIP, the PPP protocol was developed as an Internet standard. We can obtain this software for SunOS from ftp://mcsun.eu.net/network/pp/ppp-sunos4.1.pl6.tar.Z. The steps of installing the PPP software and configuring a PPP interface [10] for the fire-wall is almost identical the SLIP that had been described in the last session. Thus, we only simply specify the different individual steps:

First, copy the PPP system files to /sys.

```
# cp ppp_async.c /sys/os/ppp_ansync.c ↵
# cp ppp_if.c /sys/os/ppp_if.c ↵
# cp ppp_str.h /sys/sys/ppp_str.h ↵
# cp slcompress.c /sys/os/slcompress.c ↵
# cp slcompress.h /sys/os/slcompress.h ↵
# cp slip_var.h /sys/sys/slip_var.h ↵
# cp slip_var.h /usr/include/sys/slip_var.h ↵
```

Second, insert following lines to /sys/conf.common/files.cmn:

```
os/slcompress.c          optional    ppp ↵
os/ppp_if.c              optional    ppp ↵
os/ppp_async.c           optional    ppp ↵
```

Third, edit /sys/sun/str_conf.c, adding following three pieces of code:

```
#include "ppp.h"
---
#if     NPPP > 0
extern struct streamtab ppp_asyncinfo;
extern struct streamtab ppp_ifinfo;
#endif
---
#if     NPPP > 0
        { "pppif",      &ppp_ifinfo},
        { "pppasync",   &ppp_asyncinfo},
#endif
```

Fourth, apply the same yapt5.5c kernel patches to fix for serial problems.

Fifth, edit the kernel configuration file and rebuild the kernel. The pseudo-device statement inserted to the kernel configuration file for PPP is:

```
pseudo-device       ppp5
```

Finally, configure the PPP network interface of the fire-wall by using the ppp command, and add the default route for the fire-wall. For our example in the last session, we run the PPP protocol on the remote terminal server of the regional center, and the PPP command is liking this:

```
# ppp 192.192.6.106:192.192.6.110 /dev/ttya
  & ↵
```

Then, we can make a PPP link between the fire-wall and the regional center, and provide each network user to access TANet via the fire-wall.

2.2.3 SLIP VS. PPP

What is the best serial protocol ? To answer this question, we would like to change the "best" in terms of "right". In general, the both serial protocols could be used for solving our problem, the PPP protocol is better than the SLIP protocol. But because the SLIP was created before than PPP, thus, some of terminal servers can only support the SLIP protocol. So, we recommend that depending on the functions of the terminal server of the regional center, using PPP where you can and using SLIP where you must.

3. Configure the Fire-wall as a Mail server

In session 2. we have constructed a Unix host as a fire-wall of the small campus network, and provide the Telnet, Ftp, Archie, BBS, Gopher, and News services for each user. In addition, we must provide the E-mail service for the small campus network. This is the most important subject that we connect the small campus network to TANet. For providing the E-mail service, we should configure the fire-wall as a Mail server of the small campus network.

To configure the fire-wall as a Mail server, the first thing we should do, is to provide the DNS service on the fire-wall. There are many documents referring about how to configure a DNS system [11][12]. We won't describe it again. Because there is only one legal host (the fire-wall) connect to TANet directly and most of the High and Junior High & Preliminary Schools are lack of Unix experts to maintain the DNS server, we recommend not to run a DNS server on the fire-wall. We suggest to ask the network administrator of the regional center to

add the fire-wall address information into /etc/named.hosts and /etc/named.rev files of the DNS server of regional center as it is a host of regional center, and to run the DNS client (resolver) on the fire-wall to access the DNS server of the regional center.

Because the fire-wall connects to the regional center via a low speed dial-up telephone line. This is slower than the high speed leased-line connection. Thus, we don't think the fire-wall to send and receive E-mails to/from TANet directly is the best way. We recommend to send mails to a Mail server of the regional center as it is a Mail-Hub for the fire-wall, and receive mails from this Mail server as it is a Mail-Exchanger for the fire-wall. To do this, we should perform the following two steps:

First, ask the network administrator of the regional center to define the Mail-Exchanger for the fire-wall on the DNS server. In our example, suppose the domain name of the host 192.192.6.110 in Fig.4 is cc.nhltc.edu.tw and it is a Mail-server, the domain name of the fire-wall is dhcc.nhltc.edu.tw. Then, the network administrator should add a line into /etc/named.hosts:

```
dhcc    IN    MX  0     cc.nhltc.edu.tw
```

Next, we must modify /etc/sendmail.cf file of the fire-wall. We suggest to get the modification version by National Tsing Hua University to replace the original sendmail.cf file. This file is located at /pub/sendmail/4.1 in the ftp server net.nthu.edu.tw, and named station.cf. Use the anonymous ftp to get it. Then, we simply modify following pieces of statement of the sendmail.cf file:

```
# my local host name (fully qualified)
Djdhcc.nhltc.edu.tw
# my IP address
DN[192.192.6.106]
# my aliases
CWdhcc.nhltc.edu.tw
# my mail Hub
DHcc.nhltc.edu.tw
```

Now, each user of the small campus network can send and receive mails on the fire-wall. Each mail sent from the fire-wall will be sent to the Mail-Hub first, then delivered from the Mail-Hub. When a mail sent to the fire-wall, it will be received by the Mail-Exchange and then passed to the fire-wall. Such mail delivering method is helpful for the low speed dial-up telephone circuit problem.

4. Configure the Fire-wall as a WWW Proxy Server

WWW has recently been the most popular network application on TANet. To provide the complete services for the small campus network, we should let each network user capable to use the WWW browser, like Netscape, on any unregistered PC of the small campus network to access any WWW server world wide. This can be done by configuring the fire-wall as a WWW proxy server.

To configure the fire-wall as a WWW proxy server, the fire-wall must be a WWW server first and run the httpd daemon on it. Configuring a WWW server, the reader can refer the CERN httpd Reference Manual [13] to install the server by himself. Here, we don't describe the installation steps again. To get the server software, we can use anonymous ftp to get it from ftp://ftp.w3.org/pub/www/src/cern_httpd.tar.Z. Because we are only interesting to the proxy forwarding function rather than build a WWW server, so we don't specify how to use the HTML (HyperText Markup Language) to write a Homepage, we only simply describe how to make a WWW server as a proxy server. If the reader is interesting to HTML, he can get the HyperText Markup Language manual [14] from ftp://ftp.w3.org/pub/www/doc/html-spec.ps.Z.
To enable the proxy forwarding function, edit the /etc/httpd.conf file and add following pieces of the statement:

```
#  Set the port for Proxy to listen to
Port            80
---
#  Pass the URLs that this Proxy is willing to
 # forward.
Pass     http: *
Pass     ftp: *
Pass     gopher: *
Pass     wais: *
```

After restart the httpd daemon of the fire-wall, it can forward each of http, ftp, gopher, and wais URLs (Uniform Resource Location) request to TANet to access other WWW servers for the PCs located in the small campus network. To make this to work correctly, not only the fire-wall must be configured as a WWW proxy server but also the PC clients should make the right setup. For convenience, we use the most popular WWW browser Netscape V1.1 running on a PC client to demonstrate it. In Fig.4 the fire-wall has two IP addresses, one is legal IP

address 192.192.6.106 associated to the SLIP interface; the other is an unregistered IP address 200.1.1.10 associated to the local ethernet interface. To run Netscape, we should choose the "Options" pulldown menu item at the menu-bar, and choose the "Preferences" popup menu, then choose the "proxies" to setup proxies information. We should setup the http, ftp, gopher, and wais proxies information:

FTP Proxy:	200.1.1.10	Port: 80
Gopher Proxy:	200.1.1.10	Port: 80
HTTP Proxy:	200.1.1.10	Port: 80
WAIS Proxy:	200.1.1.10	Port: 80

Now, each PC can run the WWW browser, Netscape, to access the WWW servers that outside the small campus network via the fire-wall's proxies forwarding.

5. Conclusion

Because of the limiting space of IP addresses and more and more machines connecting to Internet, it becomes that the IP address is a very precious network resource. The entire Internet community has paid attention to this problem. There were many papers proposed for solving the lack of IP addresses problem, and stricter guidelines have been announced for the IP subnets assignment. Thus, it is more difficult to apply a lot of IP subnets on TANet than before. This is a liability in introducing TANet to High and Junior High & Preliminary Schools. If all such schools connected to TANet, each of them will need at least one IP subnet, then thousands of IP subnets are encountered. Many of them will be wasted, if each scope of the subnet is small. Thus it is a challenge to the NII policy.

In this paper, we don't introduce any new technique, instead we introduce a new concept to solve the lack of IP subnets problem. The contribution of this paper is to combine several existing techniques to form a new TANet connection scheme for small campus network without consuming any registered IP subnet. But this scheme has its own weak point, that is the connection needs recourse to a low speed dial-up telephone circuit. It is not suitable for a large scope of campus network. But for most of the High and Junior High & Preliminary Schools campus, the scope of the network is very small, and we need not to worry about it. If the scale of such a school becomes large, we suggest to use the traditional high speed lease-line connection, and reserve this dial-up line as a backup connection.

It is very important for a company to collect information from Internet in this information explosive time to increase its ability for competition. This scheme can also be applied to a small company to connect to HINET or SEEDNET. It can offer Internet resources to a small company to connect more easily and rapidly.

We had described an alternative TANet connection scheme via a fire-wall in this paper. But we only have built the fire-wall on the Unix platform. It may be considered how to build the fire-wall on other platforms, such as Window NT or OS/2. It should be useful to make our solution complete, and it will be our future work to study.

Lastly, we should emphasize that it is a pratical work scheme. The performance of this scheme seems not bad even though using the WWW service providing that the concurrent network users are not too many. Now, we have helped two high schools in Hua Lien to connect to TANet by using this scheme. We hope this experience will be also helpful for the furtherance of TANet.

References

[1] Wang, Z., and J. Crowcroft, *"A Two-Tier Address Structure for the Internet: A Solution to the Problem of Address Space Exhaustion"*, RFC 1335, University College London, May 1992.

[2] K. Siyan, *"An IP Address Extension Proposal"*, RFC 1365, September 1992.

[3] P. Robinson, *"Suggestion for New classes of IP Addresses"*, RFC 1375, November 1992.

[4] Fuller, V., T., and K. Varadhan, *"Supernetting: an Address Assignments and Aggregation Strategy"*, RFC 1338, BARRNet, cisco, Merit, OARnet, June 1992.

[5] E. Gerich, *"Guidelines for Management of IP Address Space"*, RFC 1466, Merit Network, Inc., May 1993.

[6] F. da Cruz, *"C-KERMIT 5A Configuration Information"*, Columbia University, October 1994.

[7] F. da Cruz, "*C-KERMIT 5A Installation Instructions for Unix*", Columbia University, October 1994.

[8] J. Romkey, "*Nonstandard for transmission of IP datagrams over serial lines: SLIP*", RFC 1055, June 1988.

[9] W. Simpson, "*The Point-to-Point Protocol (PPP)*", RFC 1661, July 1994.

[10] Craig Hunt, "*TCP/IP Network Administration*", O'Relly & Associates, Inc., August 1992

[11] Craig Richmond, "*Setting up a basic DNS server for a domain Revision 1.1.1*", August 1993.

[12] P. Beertema, "*Common DNS Data File Configuration Errors*", RFC 1537, October 1993.

[13] Ari Luotonen, Tim Berners-Lee, "*CERN httpd Reference Manual*", CERN, May 1994.

[14] Tim Berners-Lee, CERN; Daniel Connolly, Atrium Technology Inc., "*HyperText Markup Language*", July

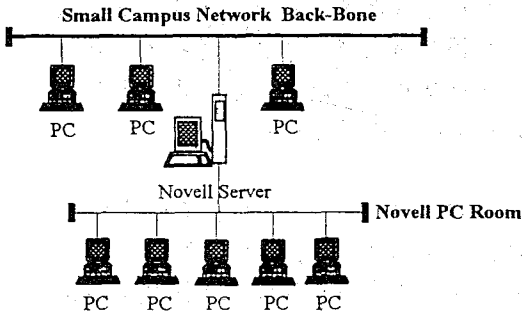

Fig. 1

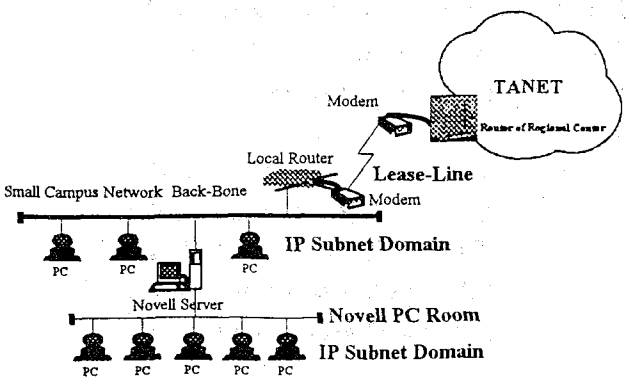

Fig. 2

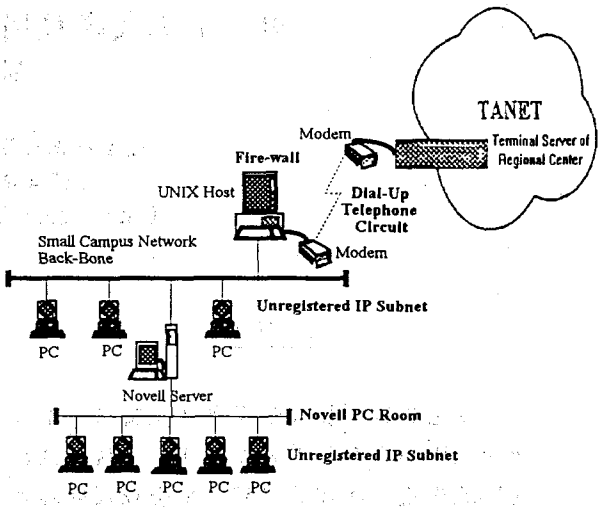

Fig. 3

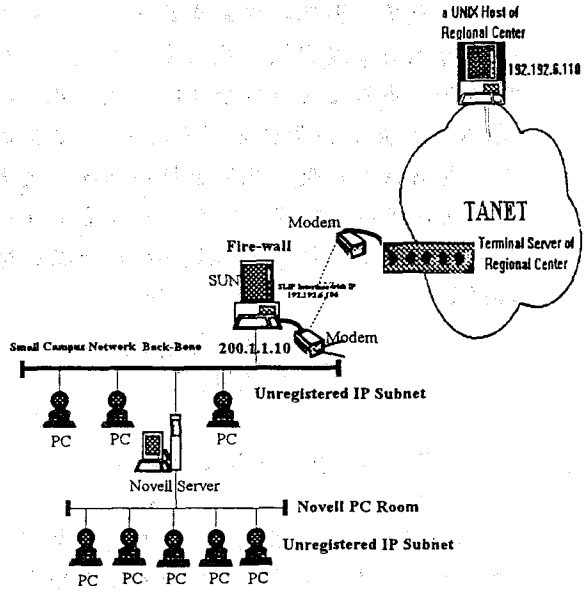

Fig. 4