Part 4: Applications

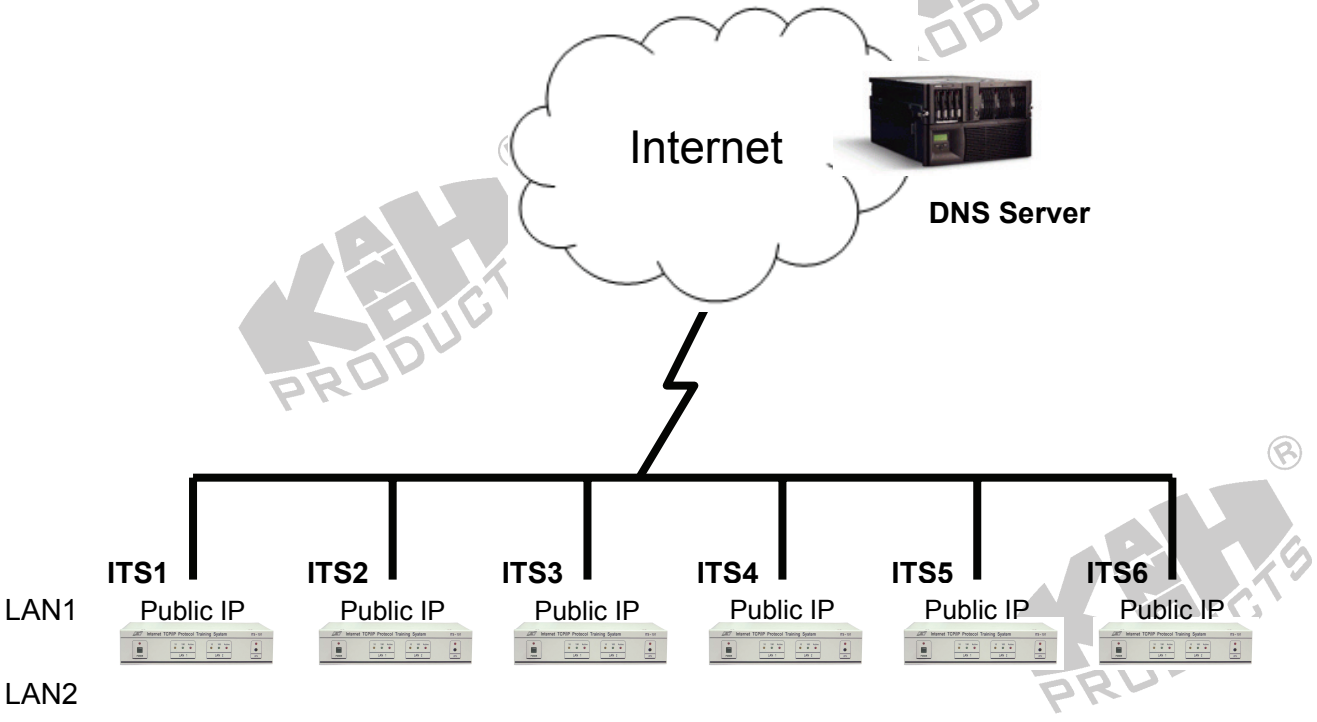
Exp 16. The Domain Name System

OBJECTIVE：To understand the applications of DNS in UDP.

BRIEF DESCRIPTION：This experiment examines the Domain Name System (DNS) that is used to resolves the domain name to an IP address. By using GUI tool, students can send DNS request messages to some sound DNS server to know what echo protocol is. Besides, by using MDDL, students can also learn how to implement a DNS client.

DURATION：3 hrs

TOPOLOGY



TECHNICAL BACKGROUND

Protocol suite:	TCP/IP
Port:	53: TCP/UDP server.

Packet Encapsulation:

MAC header	IP header	TCP/UDP header	DNS header	Data

When a client sends a domain name query request to a domain name server, the domain name server receives the query, and then checks whether or not the name lies in its authority range. If yes, it resolves the name to an IP address, and replies an answer to the client. If the name server cannot resolve the name and the client requests *recursive resolution*, the server has to contact a domain name server that can resolve the name and then returns the answer to the client. If the client requests *iterative resolution*, the name server generates a reply that specifies the name server the client should contact next to resolve the name.

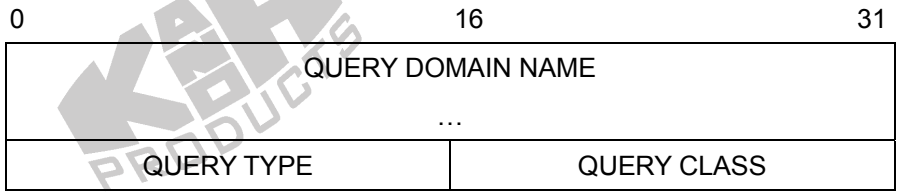
The format of DNS is shown as following figure:

0	16	31
IDENTIFACATION	PARAMETER	
NUMBER OF QUESTIONS	NUMBER OF ANSWERS	
NUMBER OF AUTHORITY	NUMBER OF ADDITIONAL	
QUESTION SECTION …		
ANSWER SECTION …		
AUTHORITY SECTION …		
ADDITIONAL INFORMATION SECTION …		

Parameter is defined as follows:

Bits of PARAMETER field	Meaning
0	Operation: 0 Query 1 Response
1-4	Query Type: 0 Standard 1 Inverse 2 Completion 1 (now obsolete) 3 Completion 2 (now obsolete)
5	Set if Answer authoritative
6	Set if Message truncated
7	Set if Recursion desired
8	Set if Recursion available
9-11	Reserved
12-15	Response Type: 0 No error 1 Format error in query 2 Server failure 3 Name does not exist

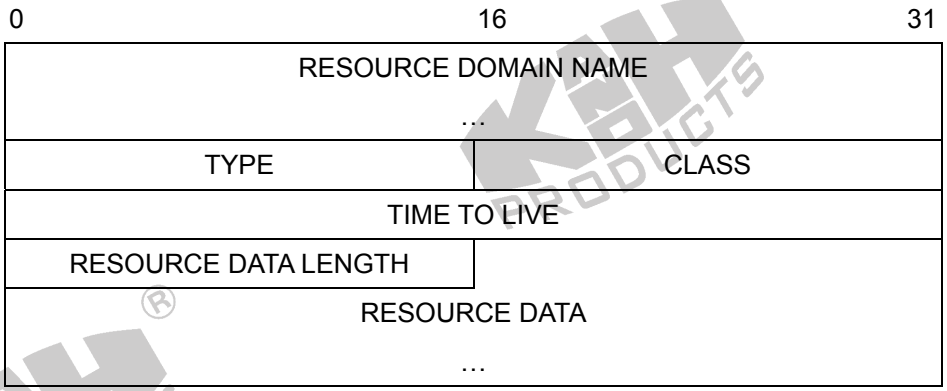
The QUESTION SECTION is defined as follows:



The *QUERY TYPE* encodes the type of the question (e.g., machine name or a mail address). The *QUERY CLASS* field allows domain names to be used for arbitrary objects. The figure shows how the domain name "*kandh.com.tw*"is represented in QUERY DOMAIN NAME:

5	k	a	n	d	h	3	c	o	m	2	t	w	0

The ANSWER SECTION, AUTHORITY SECTION and ADDITIONAL INFORMATION SECTION are defined as follows:



The RESOURCE DOMAIN NAME is encoded in a compressed format. If the top two bits of the 8-bit segment count field are 1s, the client must take the next 14 bits as an offset position in the DNS message. If the top two bits are zero, the next 6 bits specify the number of characters in the label that follow the count octet. The *TYPE* field specifies the type of the data included in the resource record; the *CLASS* field specifies the data's class. The *TIME TO LIVE* field contains a 32-bit integer that specifies how many seconds that information in this resource record can be cached. It is used by clients who have requested a name binding and may want to cache the results. The last two fields contain the results of the binding, with the *RESOURCE DATA LENGTH* field specifying the count of octets in the *RESOURCE DATA* field.

PROCEDURE

In this experiment, every ITS needs a real IP address for Internet.

Realizing Network Topology

1. Complete the network connections on HUBOX by referring to Figure 16.1.

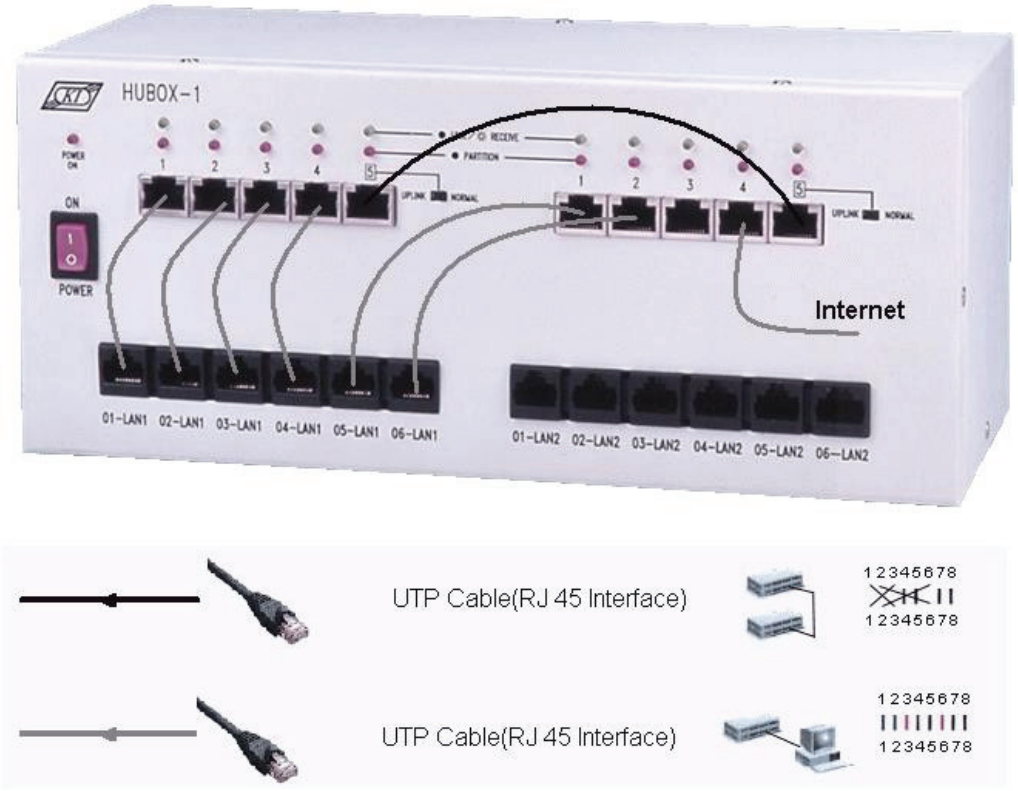


Figure 16.1

Mapping Names to Addresses

A. Setup

2. Execute **XCLIENT.BAT** to open the KCodes Network Explorer for ITS window.
3. Open the Network Message Browser window by selecting **New Memorized Message Browser** from the Listen menu.
4. In the Network Message Browser window, choose **Option** to open the **Set Message Range** dialog box as shown in Figure 16.2.
5. Click the **Add new rule** button. You need to set two rules for message browser. First type '**53**' into Remote Port, then click the **Apply** button. Secondly type "**53**" into Local

Port, then click the **Apply** again.

6. Finally click the **Set & Close** button.

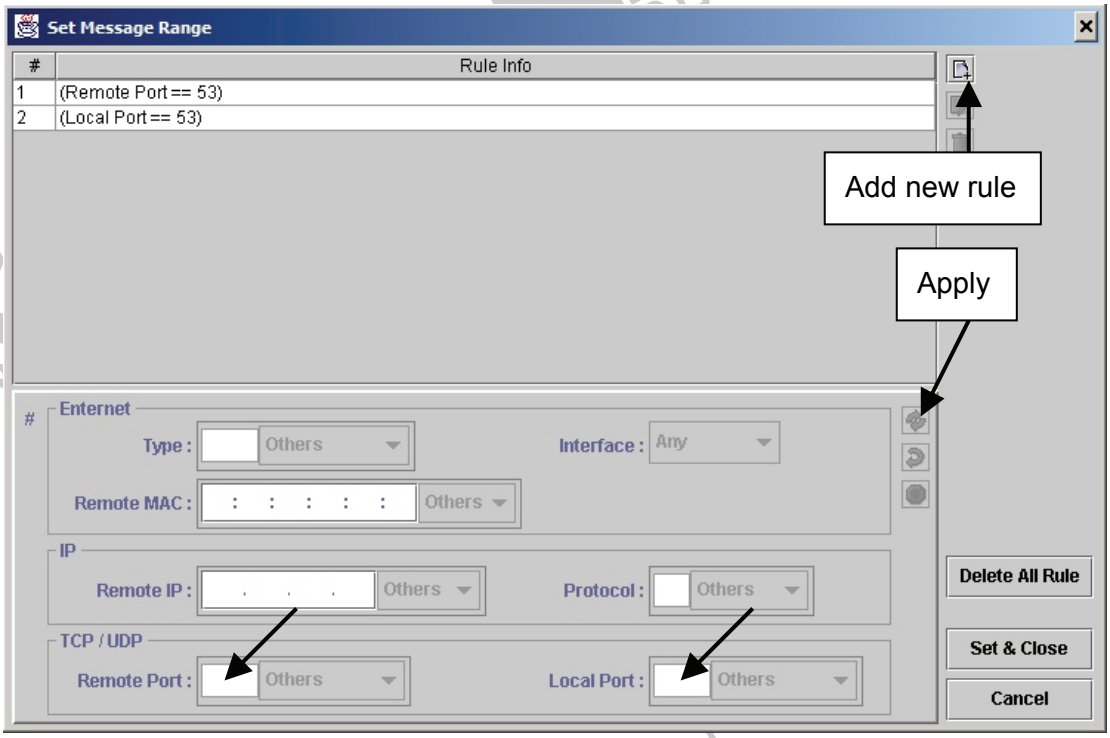


Figure 16.2

7. Open the Network Configuration dialog box by selecting **Network Configuration** from the Tool menu.

8. Type <***your Internet IP address***> into IP Address of Interface 1, set <***your Internet gateway address***> into Routing Table. For example, type "**192.168.1.223**" into IP Address of Interface 1, then enter "**192.168.1.254**" into Gateway and "**0.0.0.0**" into Destination and Mask in the Routing Table. (See Figure 16.3.)

9. Choose **Host**, and click the **Set & Close** button.

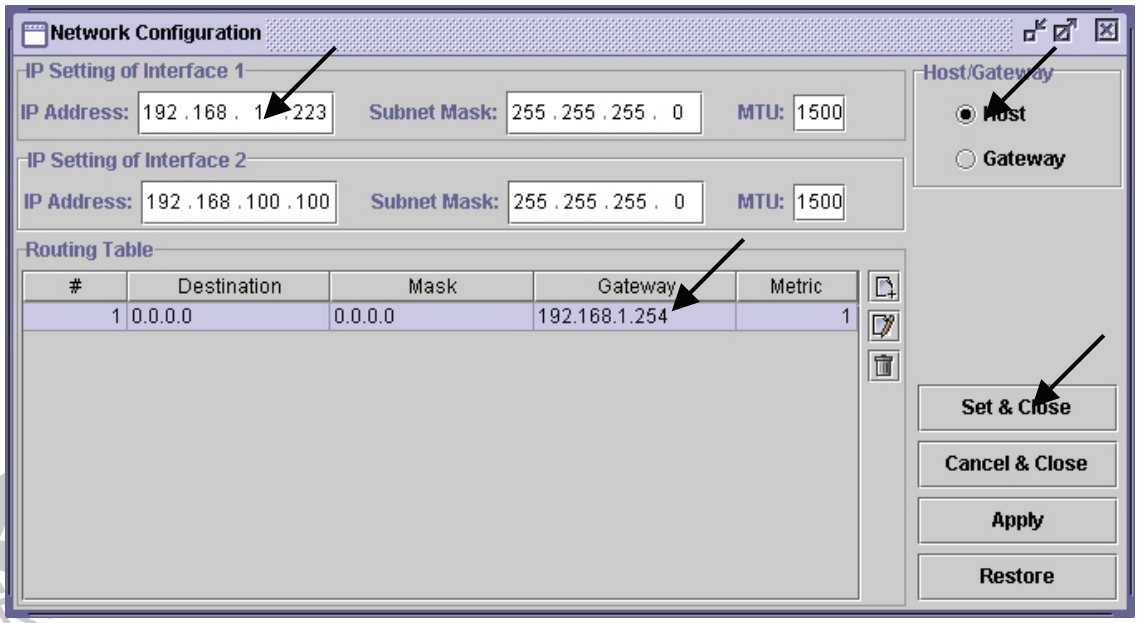


Figure 16.3

B. Sending UDP

10. Open the IP Datagram Sender by selecting **Send IP Packet** from the Send menu.
11. Type <***your Internet DNS server address***> into Destination IP Address. For example, enter "**168.95.1.1**" into Destination IP Address.
12. Enter "**kandh.com.tw**" into Data as shown in Figure 16.4.

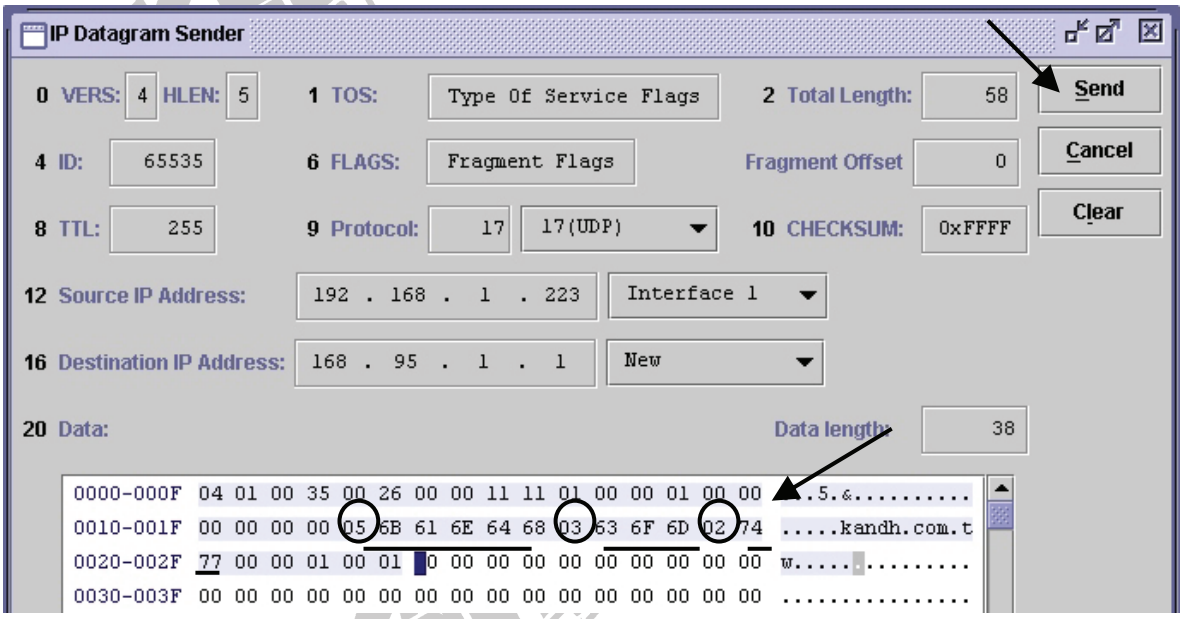


Figure 16.4

13. Finally click the **Send** button. ITS will immediately send a UDP query to query **kandh.com.tw**. You should receive a UDP datagram back as shown in Figure 16.5.

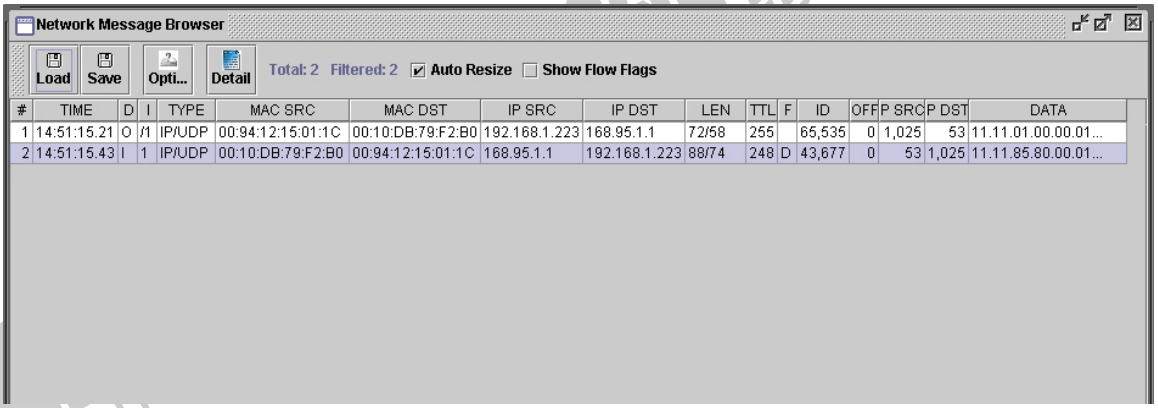


Figure 16.5

14. Select the UDP datagram and click the **Detail** button in the Network Message Browser window. You will see the detail of this UDP datagram as shown in Figure 16.6. The last 4 hex data indicate the IP address of kandh.com.tw (61.218.30.102).

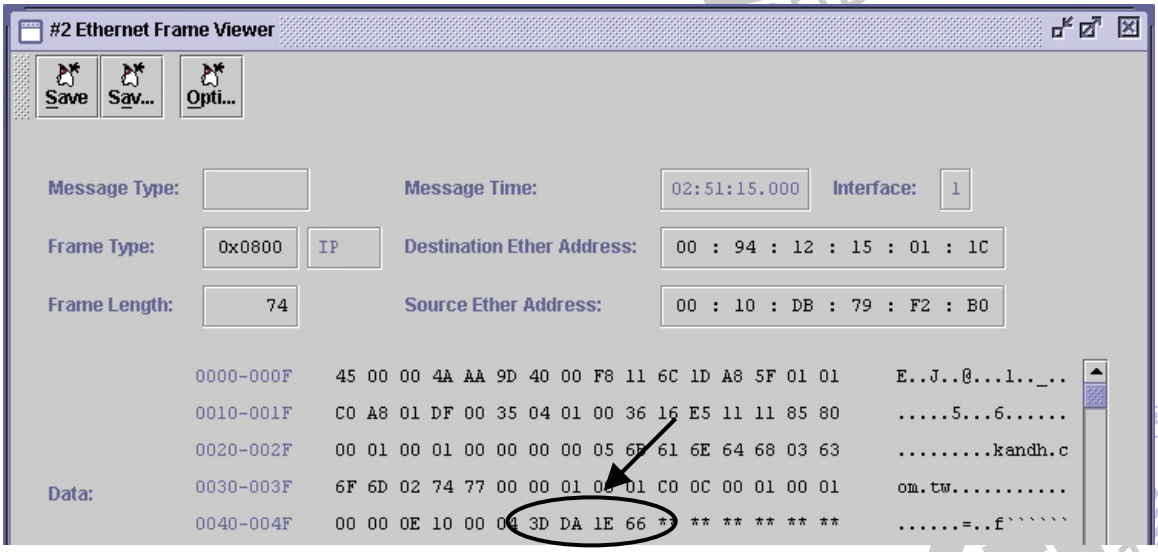


Figure 16.6

Mapping Addresses to Names

15. Open the Command Prompt window.
16. Type command **tracert 61.218.30.102**. System will query domain name of the IP address and find “kandh.com.tw” as shown in Figure 16.7.

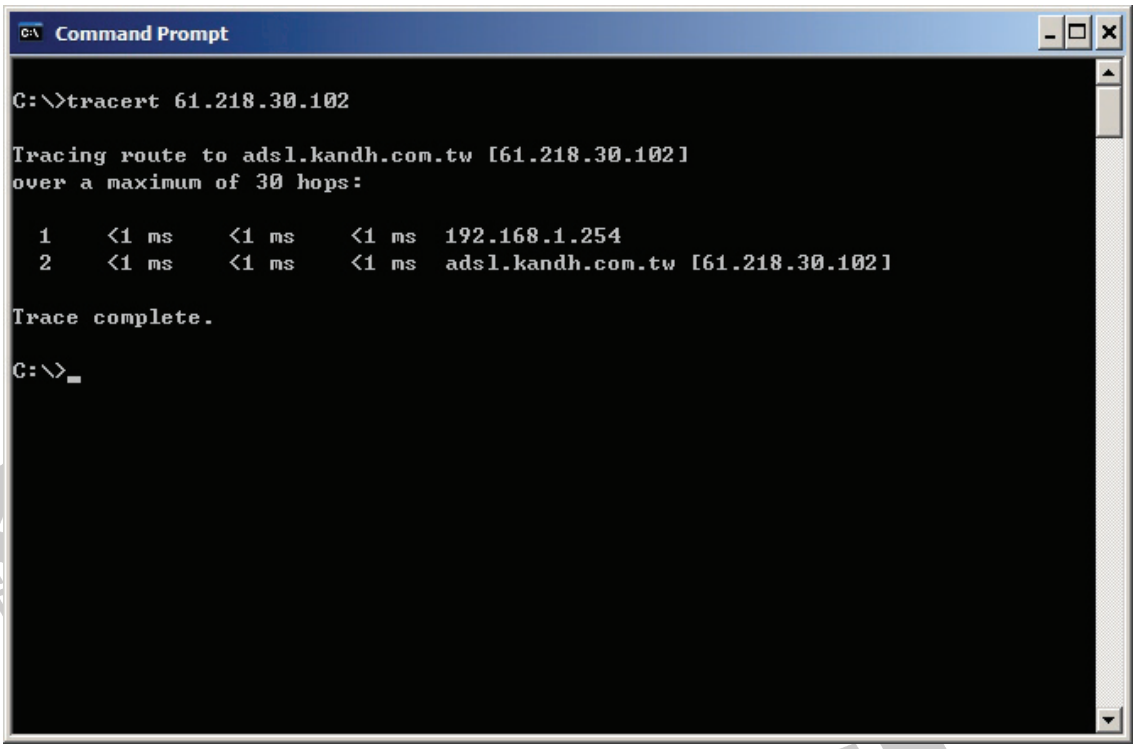


Figure 16.7

DISCUSSION

1. PC uses PING program with command “ping -a 61.218.30.102”. Observe the Network Browser and describe what happened to it.

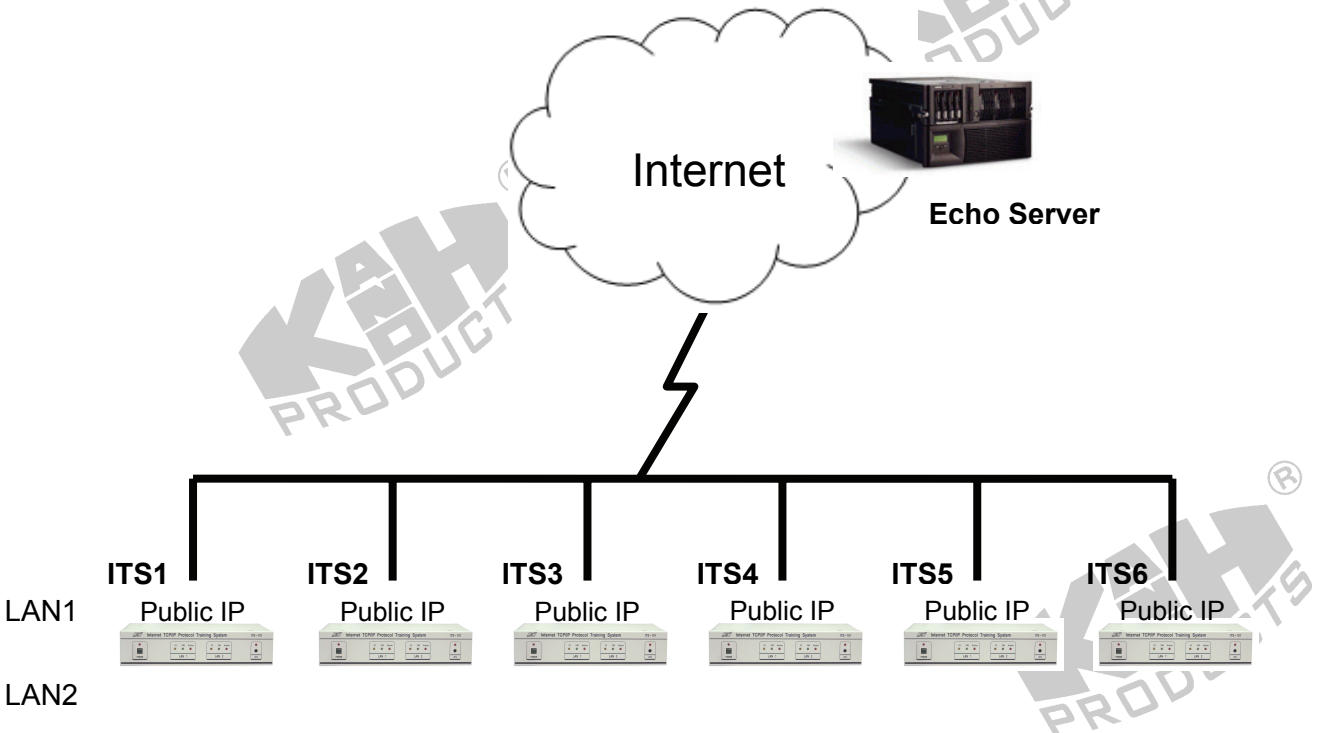
Exp 17. Echo

OBJECTIVE：To understand the applications of Echo in UDP and TCP.

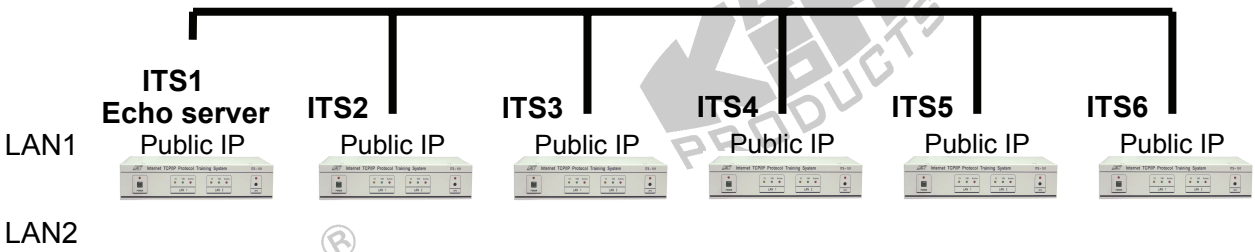
BRIEF DESCRIPTION： This experiment examines Echo Protocol that is used to get echo messages to servers. By using **TCPS** GUI tool, students can send Echo messages to some sound Echo server to know what echo protocol is. Besides, by using MDDL, students can also learn how to implement a UDP echo client and server.

DURATION: 3 hrs

TOPOLOGY A



TOPOLOGY B



TECHNICAL BACKGROUND

Protocol suite:	TCP/IP
Port:	7: TCP/UDP server.

Packet Encapsulation:

MAC header	IP header	TCP/UDP header	Echo header	Data

Echo protocol is defined in RFC 862 with the following definitions:

1. TCP Based Echo Service

   One echo service is defined as a connection based application on TCP.A server listens for TCP connections on TCP port 7. Once a connection is established any data received is sent back. This continues until the calling user terminates the connection.

2. UFC Based Echo Service

   Another echo service is defined as a datagram based application on UFC. A server listens for UFC datagrams on UFC port 7. When a datagram is received, the data from it is sent back in an answering datagram.

PROCEDURE

In this experiment, every ITS needs a real IP address for Internet.

Realizing Network Topology

1. Complete the network connections on HUBOX by referring to Figure 17.1.

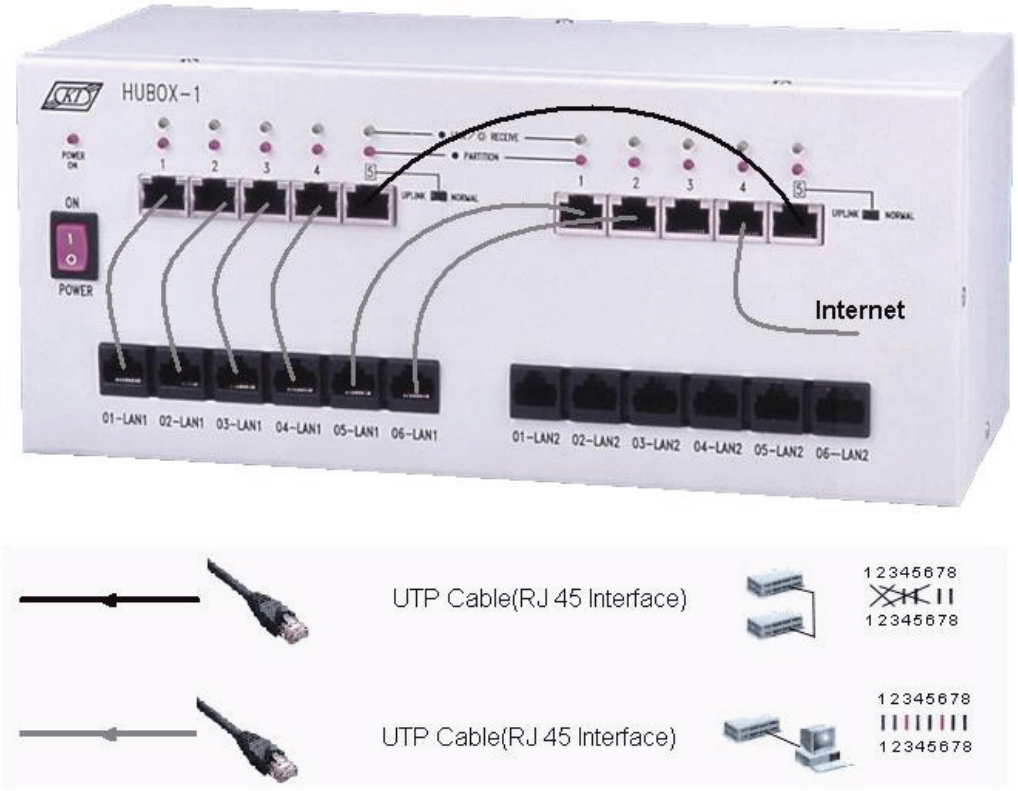


Figure 17.1

Running Echo Client to Internet Echo

A. Setup

2. Execute **XCLIENT.BAT** to open the KCodes Network Explorer for ITS window.
3. Open the Network Message Browser window by selecting **New Memorized Message Browser** from the Listen menu.
4. In the Network Message Browser window, select **Option** to open the Set Message Range dialog box.
5. Click the **Add new rule** button. You need to set two rules for message browser. First type "**7**" into Remote Port, then click the **Apply** button. Secondly type "**7**" into Local Port,

then click the **Apply** again. (See Figure 17.2.)

6. Finally click the **Set & Close** button.

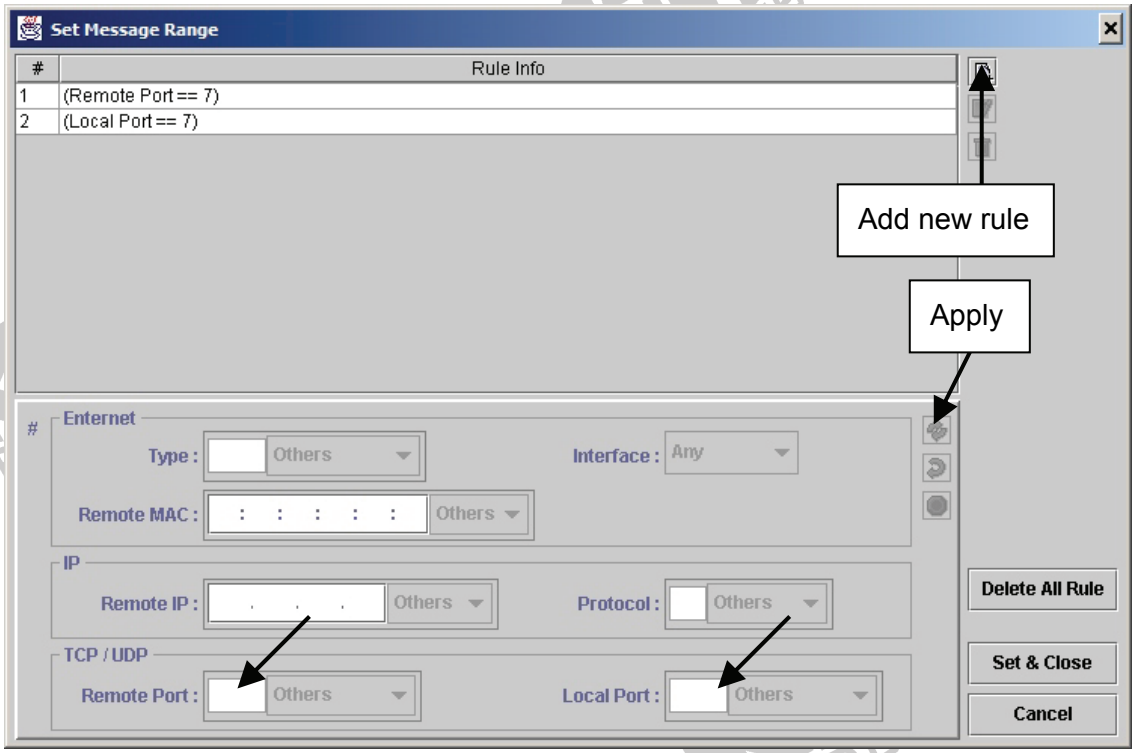


Figure 17.2

7. Open the Network Configuration dialog box by selecting **Network Configuration** from the Tool menu.

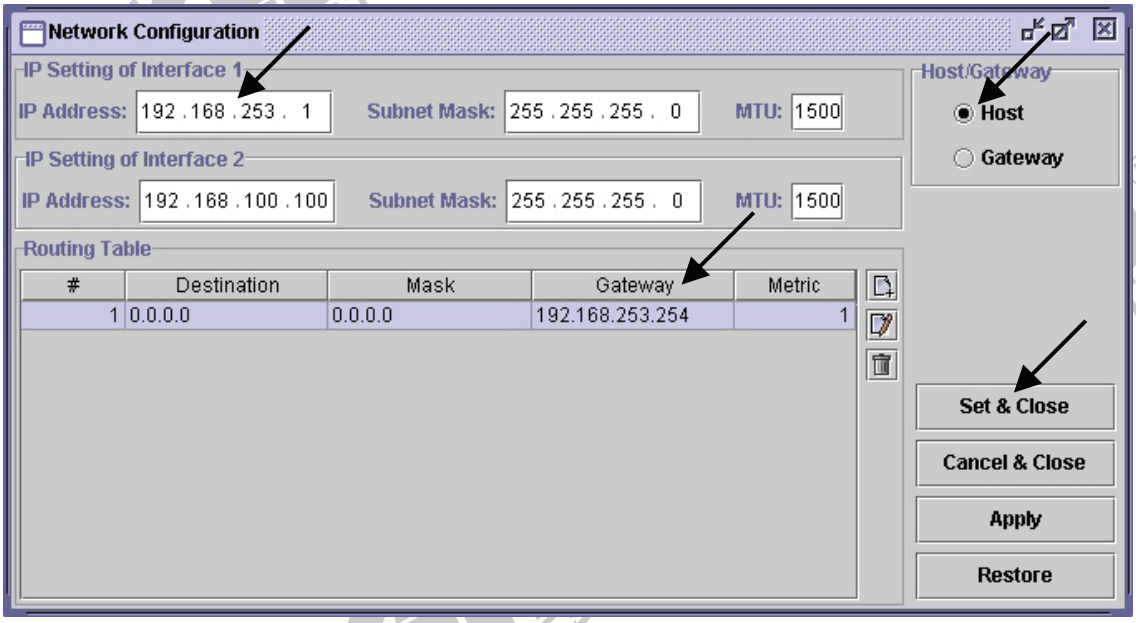


Figure 17.3

8. Type <***your Internet IP address***> into IP Address of Interface 1, and enter <***your Internet gateway address***> into Routing Table. For example, type "**192.168.253.1**" into IP Address of Interface 1, then enter "**192.168.253.254**" into Gateway and "**0.0.0.0**" into Destination and Mask in the Routing Table. (See Figure 17.3.)
9. Choose **Host** and click the **Set & Close** button.

B. Run Echo Client

10. Click **New TCP Session** from the TCP menu to open the New TCP Session dialog box.
11. Select **System Default TCP**. Type <***your Echo server IP address***> into Destination IP Address, choose **ECHO (7)** from Destination Port. For example, type "**203.149.174.99**" into Destination IP Address.
12. Click the **Connect** button.
13. Once Echo Server connected, enter "**ABC**" into edit box and click the **Send** button. Your ITS should receive "**ABC**" back immediately.

Echo in UFC

In this training, we let ITS1 act as Echo Server. Other ITS can try to send a UFC Echo datagram to ITS1.

1. Complete the network connections on HUBOX by referring to Figure 17.4.

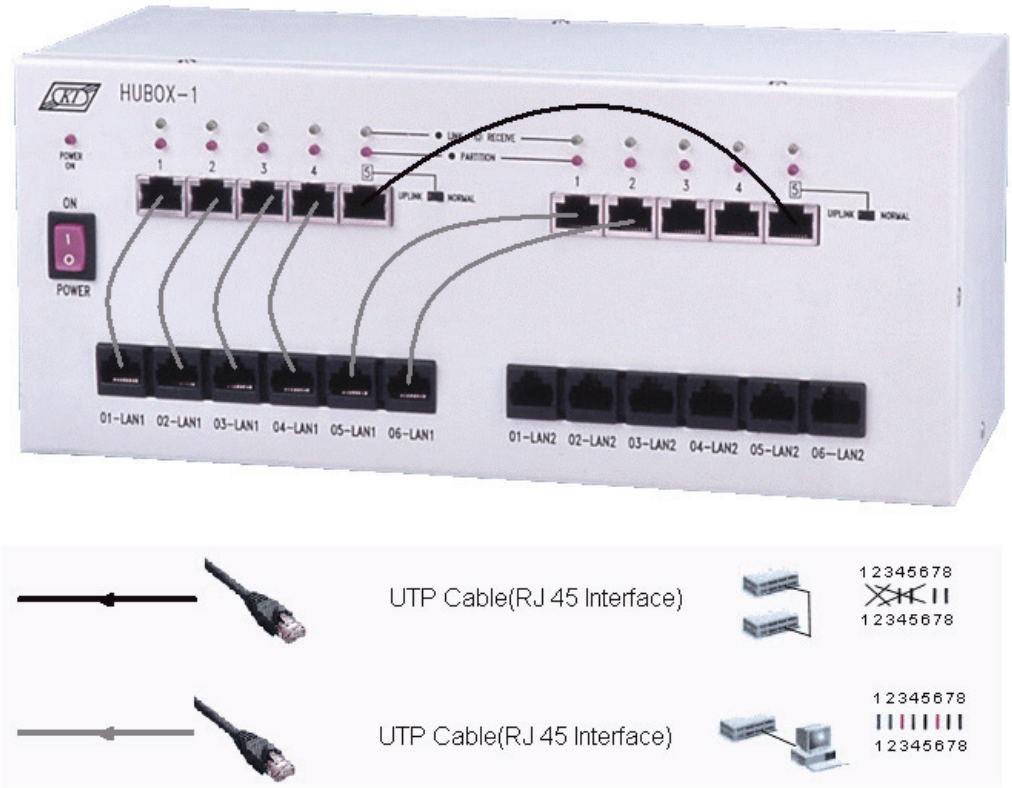


Figure 17.4

ITS1(Echo Server)

2. Open the MDDL Editor by selecting **MDDL Reactor Panel** from the Reactor menu.
3. Click the **Load** button. Open the file C: \XClient \Data \Mddl \Tutorial \Ex17 \UFCEchoServer.mddl, and click the **UpId** button.

ITS2 thru ITS6

4. Open the MDDL Editor by selecting **MDDL Reactor Panel** from the Reactor menu.
5. Click the **Load** button. Open the file C: \XClient \Data \Mddl \Tutorial \Ex17 \UFCEchoClient.mddl, and click the **UpId** button.
6. Open the New TCP Session dialog box by selecting **New TCP Session** from the TCP menu.

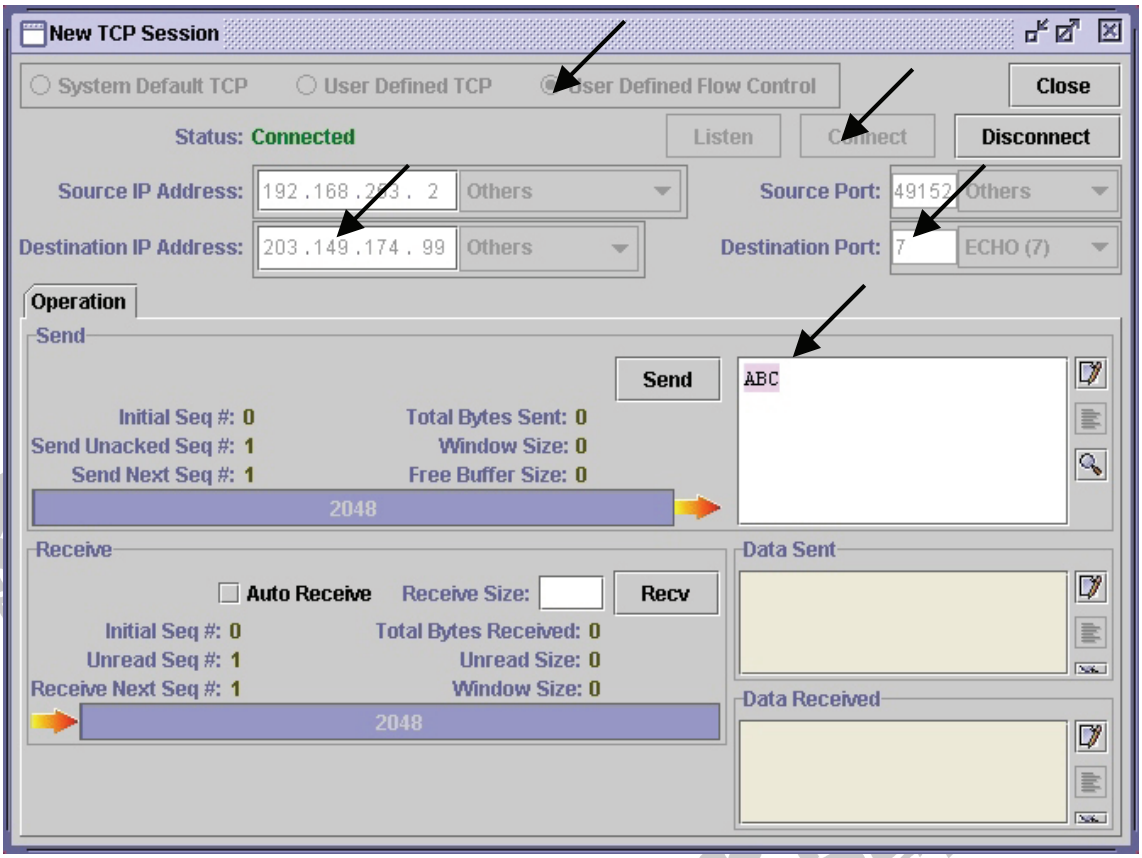


Figure 17.5

7. Select **User Defined Flow Control**. Type <***ITS1's IP address***> into Destination IP Address, choose **ECHO (7)** from Destination Port. For example, type "**203.149.174.99**'" into Destination IP Address as shown in Figure 17.5.
8. Click the **Connect** button. After the connection is successful, enter "**ABC**" into edit box and click the **Send** button.
9. Figure 17.6 shows that ITS2 sends "ABC" to ITS1. You can see it in the Data Sent mailbox.

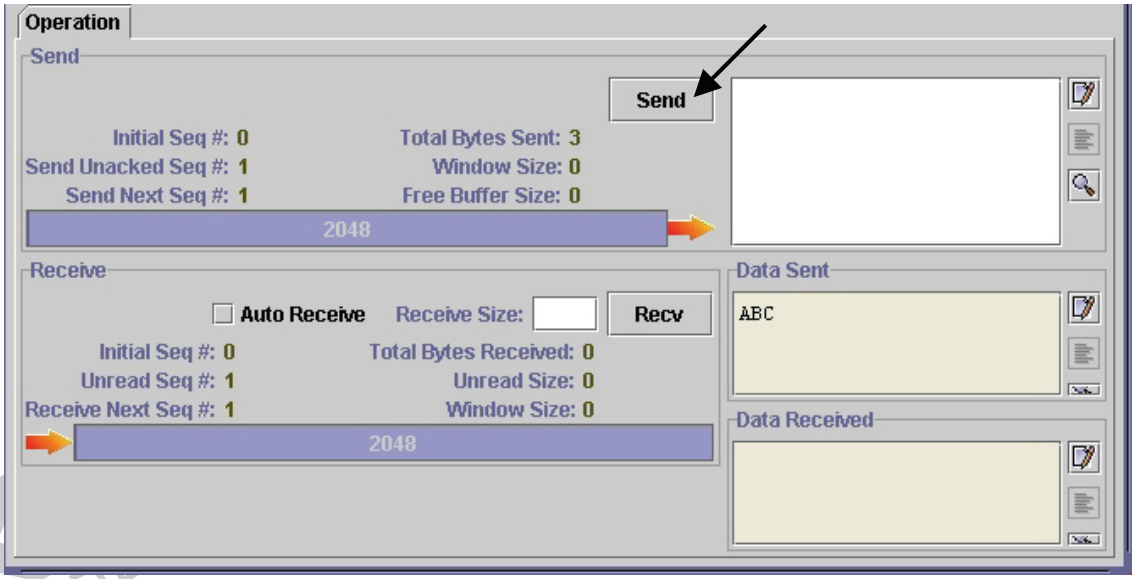


Figure 17.6

10. Click the **Recv** button. You will get ITS1 Echo Reply in the Data Received mailbox (see Figure 17.7).

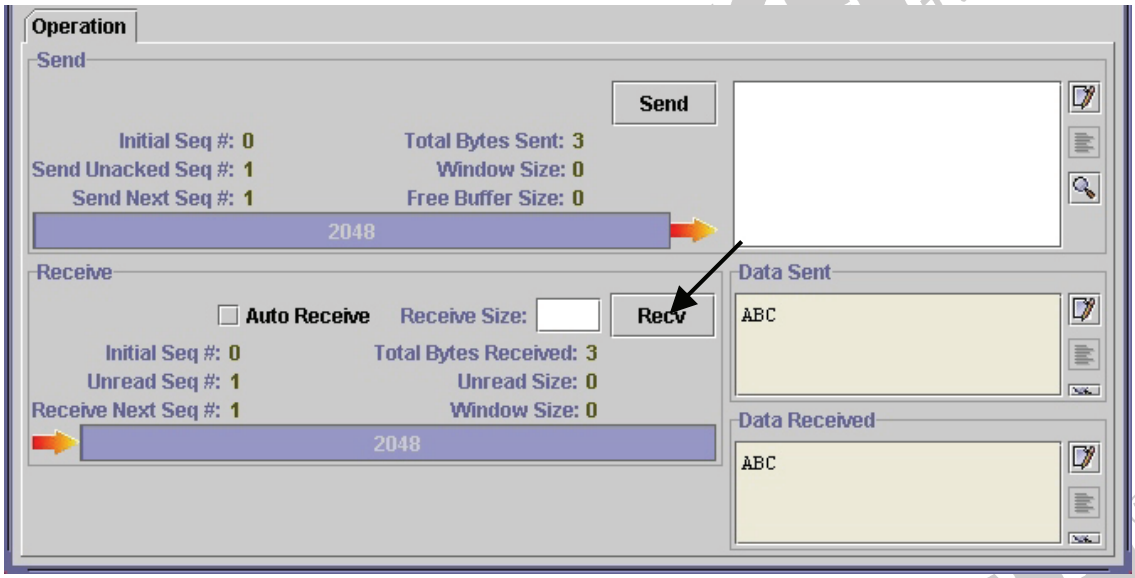


Figure 17.7

REACTOR PROGRAM

1. **UFCEchoClient.mddl**

```
VAR6.UCB_SOCKET_ID       = 5                                    ;
VAR6.UCB_STATE           = CNST_UFC_STATE_CLOSED    ;

SERVICE_UFC_OPEN
{
    IF( VAR6.UCB_STATE != CNST_UFC_STATE_CLOSED )
        RETURN;

    VAR6.UCB_STATE                                  = CNST_UFC_STATE_ESTABLISHED;

    IF(PARA_IPADDR_SRC() == CNST_IP_ADDR_BROADCAST)
        VAR6.UCB_IP_ADDRSRC = MYIP(1);
    ELSE
        VAR6.UCB_IP_ADDRSRC = PARA_IPADDR_SRC();

    VAR6.UCB_IP_ADDRDST                             = PARA_IPADDR_DST()             ;

    IF(PARA_PORT_SRC() == 0W)
        VAR6.UCB_PORTSRC                            = 49152W;
    ELSE
        VAR6.UCB_PORTSRC                            = PARA_PORT_SRC()               ;

    VAR6.UCB_PORTDST                                = PARA_PORT_DST()               ;

    RETVAL_SOCKET_ID     = VAR6.UCB_SOCKET_ID;
    RETVAL_IPADDR_SRC    = VAR6.UCB_IP_ADDRSRC;
    RETVAL_PORT_SRC      = VAR6.UCB_PORTSRC;
    RETVAL_ERRORCODE     = CNST_UFC_NO_ERROR;
}

SERVICE_UFC_LISTEN
{
    IF( VAR6.UCB_STATE != CNST_UFC_STATE_CLOSED )
        RETURN;

    VAR6.UCB_STATE           = CNST_UFC_STATE_LISTEN ;
    VAR6.UCB_IP_ADDRSRC      = PARA_IPADDR_SRC()     ;
    VAR6.UCB_IP_ADDRDST      = PARA_IPADDR_DST()     ;
    VAR6.UCB_PORTSRC         = PARA_PORT_SRC()       ;
    VAR6.UCB_PORTDST         = PARA_PORT_DST()       ;

    WAIT_SIGNAL VAR6.UCB_SOCKET_ID;

    RETVAL_SOCKET_ID     = VAR6.UCB_SOCKET_ID;
    RETVAL_IPADDR_SRC    = VAR6.UCB_IP_ADDRSRC;
    RETVAL_IPADDR_DST    = VAR6.UCB_IP_ADDRDST;
    RETVAL_PORT_DST      = VAR6.UCB_PORTDST;
    RETVAL_ERRORCODE     = CNST_UFC_NO_ERROR;
}

SERVICE_UFC_CLOSE
{
    IF( VAR6.UCB_STATE != CNST_UFC_STATE_ESTABLISHED )
        RETURN;

    VAR6.UCB_STATE                                  = CNST_UFC_STATE_CLOSED;
```

```
    GENERATE_DISCONNECTED(1, VAR6.UCB_SOCKET_ID);
}

SERVICE_UFC_SEND
{
    VAR7.UFC_PSEUDO_IP_ADDRSRC              = VAR6.UCB_IP_ADDRSRC    ;
    VAR7.UFC_PSEUDO_IP_ADDRDST              = VAR6.UCB_IP_ADDRDST    ;
    VAR7.UFC_PSEUDO_ZERO                    = 0                      ;
    VAR7.UFC_PSEUDO_PROT                    = CNST_IP_PROT_UFC       ;
    VAR7.UFC_PSEUDO_LEN                     = 8W    ;

    VAR7.UFC_PSEUDO_DATA.UFC_PORTSRC        = VAR6.UCB_PORTSRC   ;
    VAR7.UFC_PSEUDO_DATA.UFC_PORTDST        = VAR6.UCB_PORTDST    ;
    VAR7.UFC_PSEUDO_DATA.UFC_LEN            = PARA_SOCKET_BUFFER_LEN() + 8 ;
    VAR7.UFC_PSEUDO_DATA.UFC_CHKSUM         = 0W ;

    VAR7.UFC_PSEUDO_DATA.UFC_DATA           = PARA_DATA();

    VAR7.UFC_PSEUDO_DATA.UFC_CHKSUM         = CHECKSUM(VAR7[0, 12 +
    VAR7.UFC_PSEUDO_LEN + PARA_SOCKET_BUFFER_LEN() - 1 ])    ;

    SEND_OUT_IP WITH_DATA
    {
        T.IP_PROT                           = CNST_IP_PROT_UFC ,
        T.IP_ADDRDST                        = VAR6.UCB_IP_ADDRDST   ,
        T.IP_DATA       = VAR7.UFC_PSEUDO_DATA.[0, VAR7.UFC_PSEUDO_LEN +
        PARA_SOCKET_BUFFER_LEN() - 1]
    }

    RETVAL_DATA_LEN = PARA_SOCKET_BUFFER_LEN();
    RETVAL_DATA = PARA_DATA();
}

SERVICE_UFC_RECEIVE
{

    RETVAL_DATA_LEN = 0;
    FOR_EVERY_ELEMENT_IN_POOL 30
    {
        RETVAL_DATA[ RETVAL_DATA_LEN, ] = PE[0, ];
        RETVAL_DATA_LEN += LENGTH(PE);
        REMOVE_CURRENT_POOL_ELEMENT;
    }
}


IP_IN_HANDLER
{

 IF(S.IP_PROT!=CNST_IP_PROT_UFC)
        RETURN;

    IF( VAR6.UCB_STATE == CNST_UFC_STATE_ESTABLISHED   )
    {
        ADD_TO_POOL 30 WITH_DATA
        {
            TARGET = S.IP_DATA.UFC_DATA
        }
```

```
            GENERATE_SEND_BUFFER_PARAMETERS_CHANGED(VAR6.UCB_SOCKET_ID, 0, 0, 0,
             0);
        }
DISCARD_MESSAGE;
}
```

2. **UFCEchoServer.mddl**

```
//UDP ECHO SERVER listen on port 7

IP_IN_HANDLER
{
        IF(S.IP_DATA.[2, 3]==7W)
        {
                SEND_OUT_IP WITH_DATA
                {
                        T                       = S                     ,
                        T.IP_ADDRSRC            = S.IP_ADDRDST      ,
                        T.IP_ADDRDST            = S.IP_ADDRSRC      ,
                        T.IP_TTL                = 0XFF,
                        T.IP_DATA.[0, 1]      = 7W ,
                        T.IP_DATA.[2,3]       = S.[20,21] ,
                        T.IP_PROT               = CNST_IP_PROT_UFC      ,
                        T.IP_HEADERCHKSUM     = {0, 0}                  ,
                        T.IP_HEADERCHKSUM     = CHECKSUM(T.IP_HEADER)
                }
        }
DISCARD_MESSAGE ;
}
```

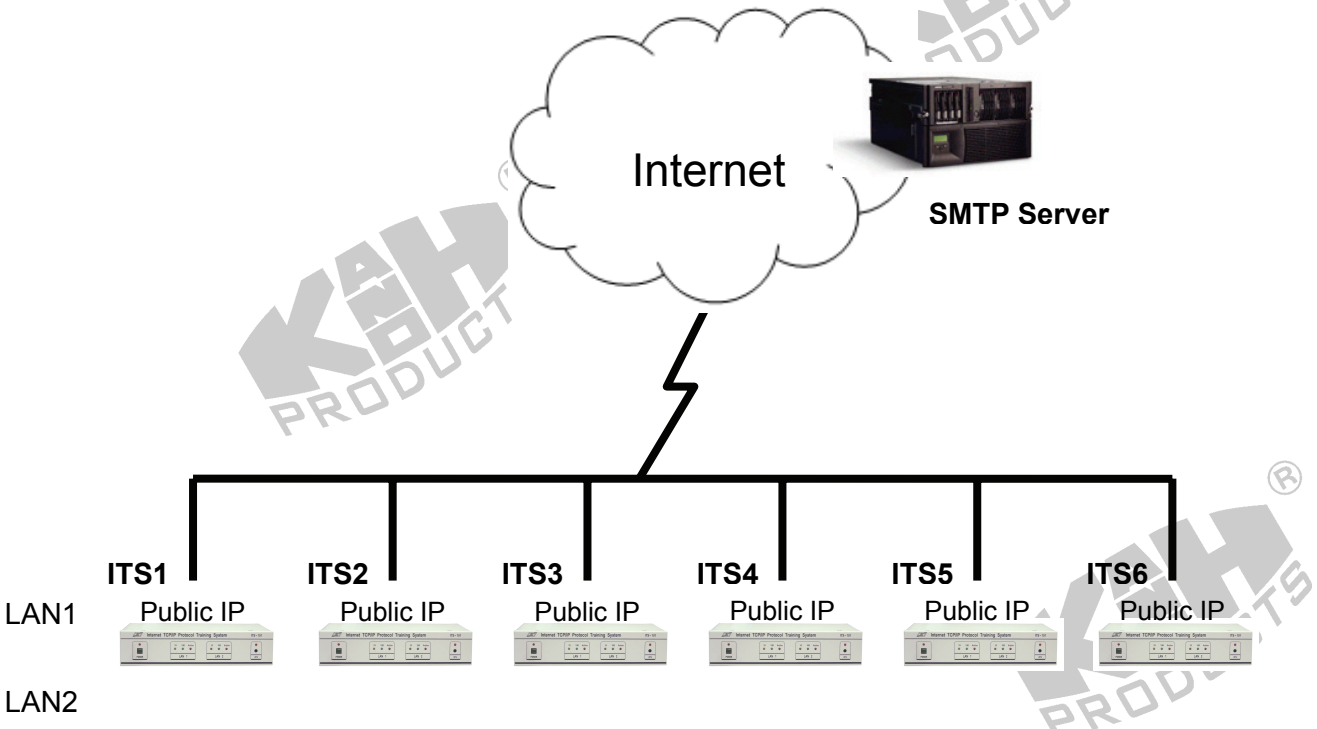
Exp 18. SMTP Client

OBJECTIVE：To understand the applications of SMTP in TCP.

BRIEF DESCRIPTION： This experiment examines Simple Mail Transfer Protocol (SMTP) that is used to clients to get mail to servers. By using **TCPS** GUI tool, students can send SMTP commands to some sound SMTP server to know what SMTP protocol is. Besides, students can also learn how to implement a HTTP client.

DURATION: 3 hrs

TOPOLOGY



TECHNICAL BACKGROUND

Protocol suite:	TCP/IP
Port:	25: TCP server.

Packet Encapsulation：

MAC header	IP header	TCP header	SMTP header	Data

SMTP is outlined in RFC 821 and is used for clients to get mail to servers. When you send an email, your mail client is connecting to an SMTP server, using SMTP protocol to instruct SMTP server to deliver the mail. SMTP clients establish a TCP connection to port 25 of SMTP server which accepts incoming connections and copies messages from them into the appropriate mailboxes. If a message cannot be delivered, an error report is returned to the sender.

PROCEDURE

In this experiment, every ITS needs a real IP address for Internet.

Realizing Network Topology

1. Complete the network connections on HUBOX by referring to Figure 18.1.

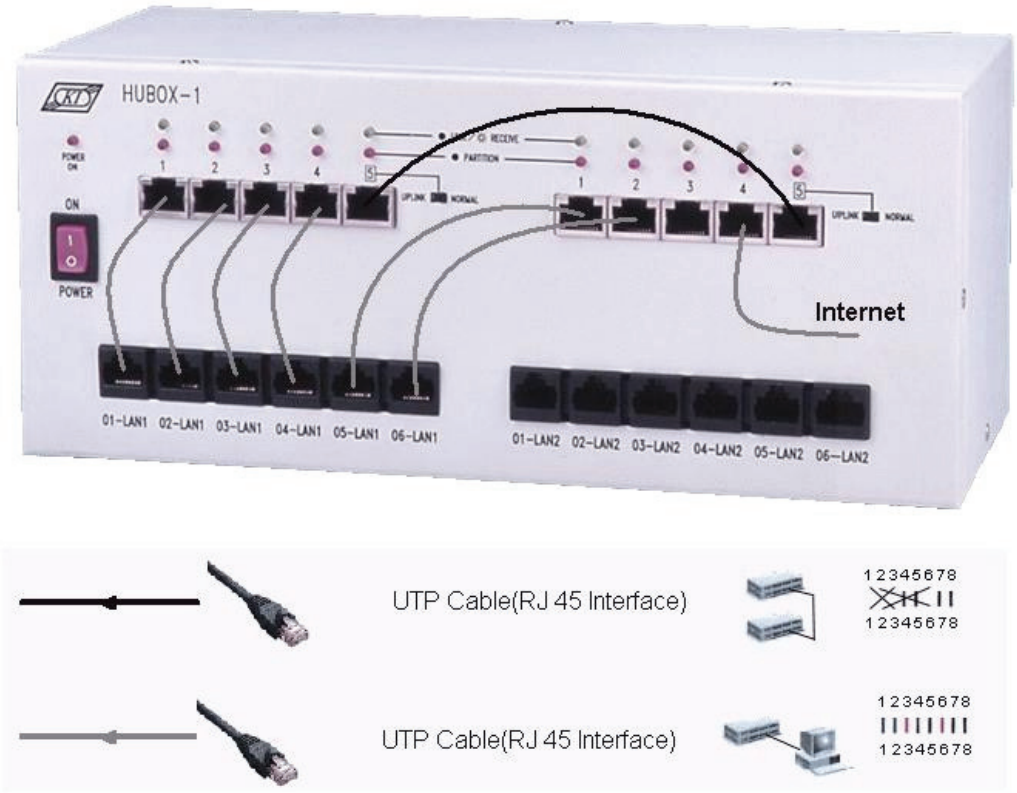


Figure 18.1

SMTP Procedure

A. Setup

2. Execute **XCLIENT.BAT** to open the KCodes Network Explorer for ITS window.
3. Open **Network Message Browser** by selecting **New Memorized Message Browser** from **Listen** menu.
4. Choose **Option** from the Network Message Browser to open the Set Message Range as shown in Figure 18.2.
5. Click the **Add new rule** button. You need to set two rules for message browser. First type "**25**" into Remote Port, then click the **Apply** button. Secondly type "**25**" into Local

Port, then click the **Apply** button again.

6. Finally click the **Set & Close** button.

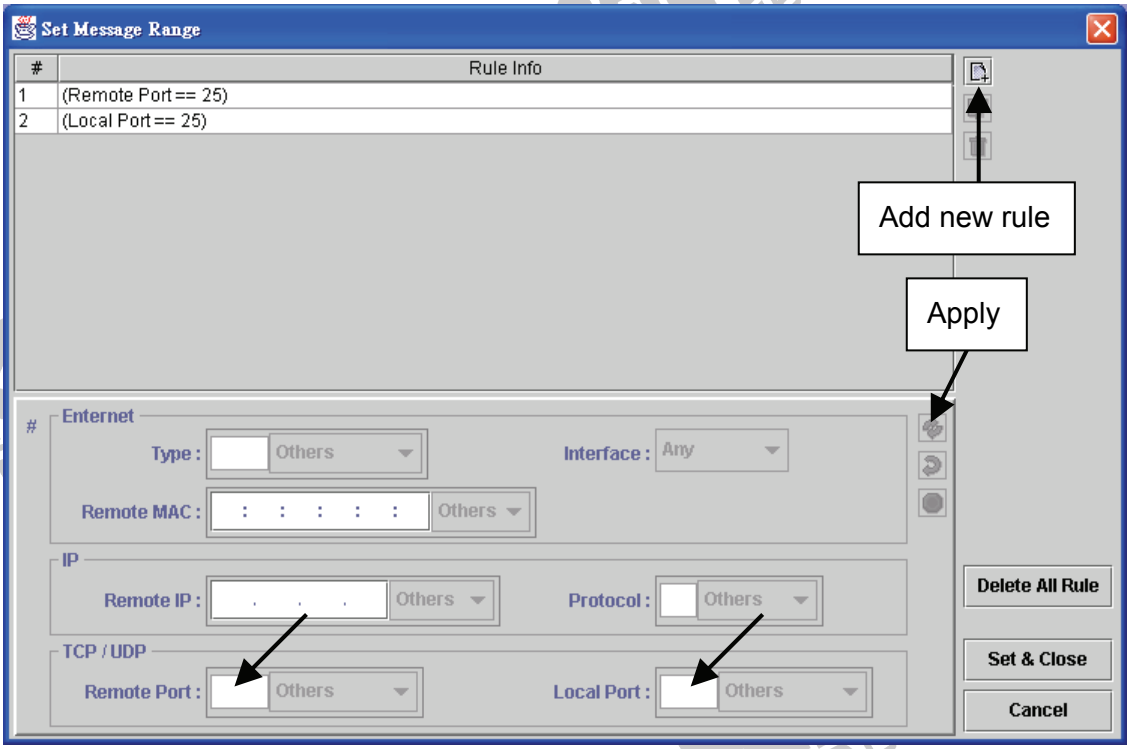


Figure 18.2

7. Select **Network Configuration** from the Tool menu to open the Network Configuration dialog box.

8. Type <***your Internet IP address***> into IP Address of Interface 1, enter <***your Internet gateway address***> into Routing Table. For example, type "**192.168.1.223**" into IP Address of Interface 1, and enter "**192.168.1.254**" into Gateway and "**0.0.0.0**" into Destination and Mask in the Routing Table. (See Figure 18.3.)

9. Choose **Host**, and click the **Set & Close** button.

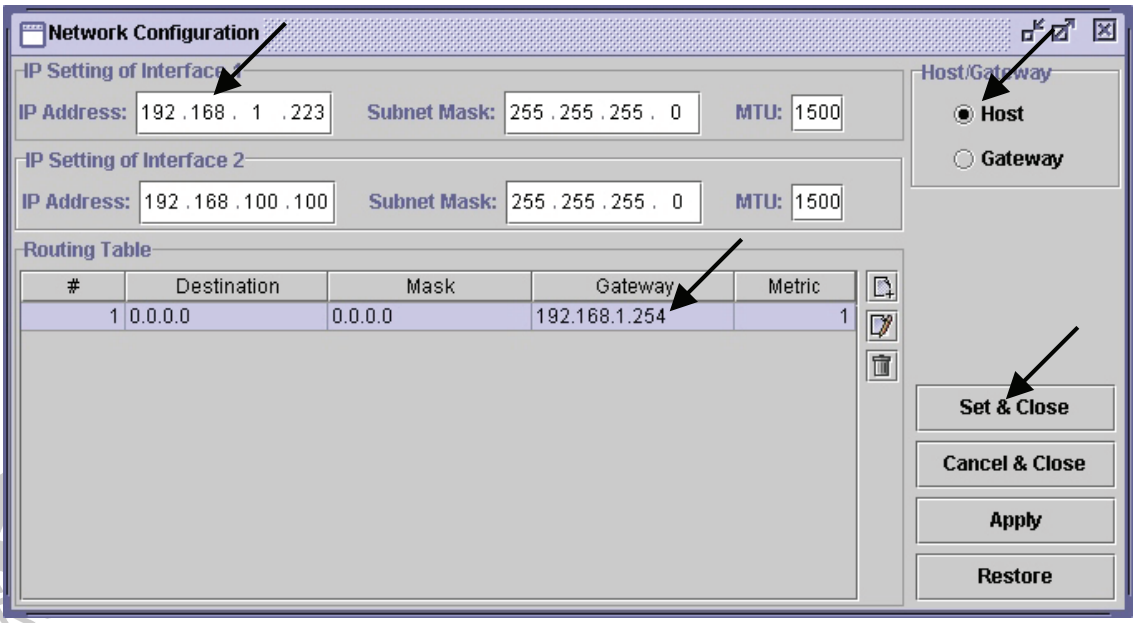


Figure 18.3

B. Sending Mail

10. Open the New TCP Session dialog box by selecting **New TCP Session** from the TCP menu.
11. Select **System Default TCP**. Type <***your SMTP server IP address***> into Destination IP Address, choose **SMTP (25)** from Destination Port. For example, type "**61.218.30.102**" into Destination IP Address.
12. Check **Auto Receive**, then click the **Connect** button. You will receive SMTP server message as shown in Data Received mailbox. (See Figure 18.4.)

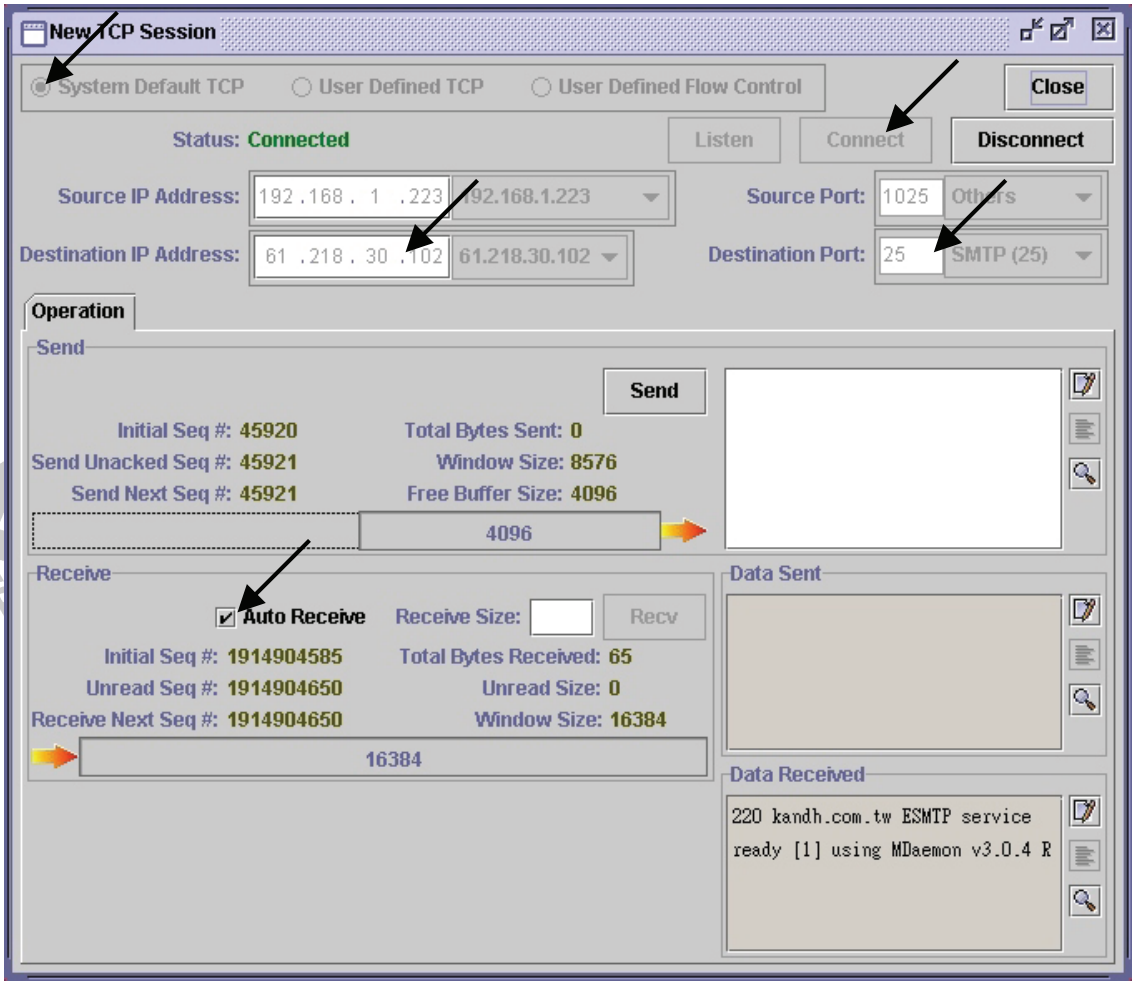


Figure 18.4

13. In the edit box, type the following commands and sentences to send an email.

    Send: **HELO** ↲

    Read: 250 kandh.com.tw HELLO, pleased to meet you

    Send: **MAIL FROM: user@mydomain.com** ↲

    Read: 250 2.1.0 user@mydomain.com... Sender ok

    Send: **RCPT TO: recipient@mail.recipient-domain.com** ↲

    Read: 250 2.1.5 recipient@ mail.recipient-domain.com... Recipient ok

    Send: **DATA** ↲

    Read: 354 Enter mail, end with <CRLF>.<CRLF>

    Send: **SUBJECT: This is a test email** ↲

    Send: **DATE: dd/mm/yy** ↲

Send: **FROM: user** ↲

Send: **TO: recipient** ↲

Send: **This is to demonstrate it as an example.** ↲

Send: **.** ↲

Read: 250 OK, message saved

Send: **QUIT** ↲

Read: 221 See ya in cyberspace

DISCUSSION

1. Describe the way how to get email send by SMTP Client.
2. How to send an mail from ITS with “Content” ?

Hint:

```
DATA SENT :
helo
mail from:<hugo>
rcpt to:<hugo@kandh.com.tw>
Data
From: hugo
To: hugo
Subject: HI
Date: Web,29 Mar 2006 14:21:49 +0800
Content-Type: text/plain;
        format=flowed;
        charset=”US-ACSII”;
        reply-type=original
Content-Transfer-Encoding: 7bit

TEST

.
quit
```

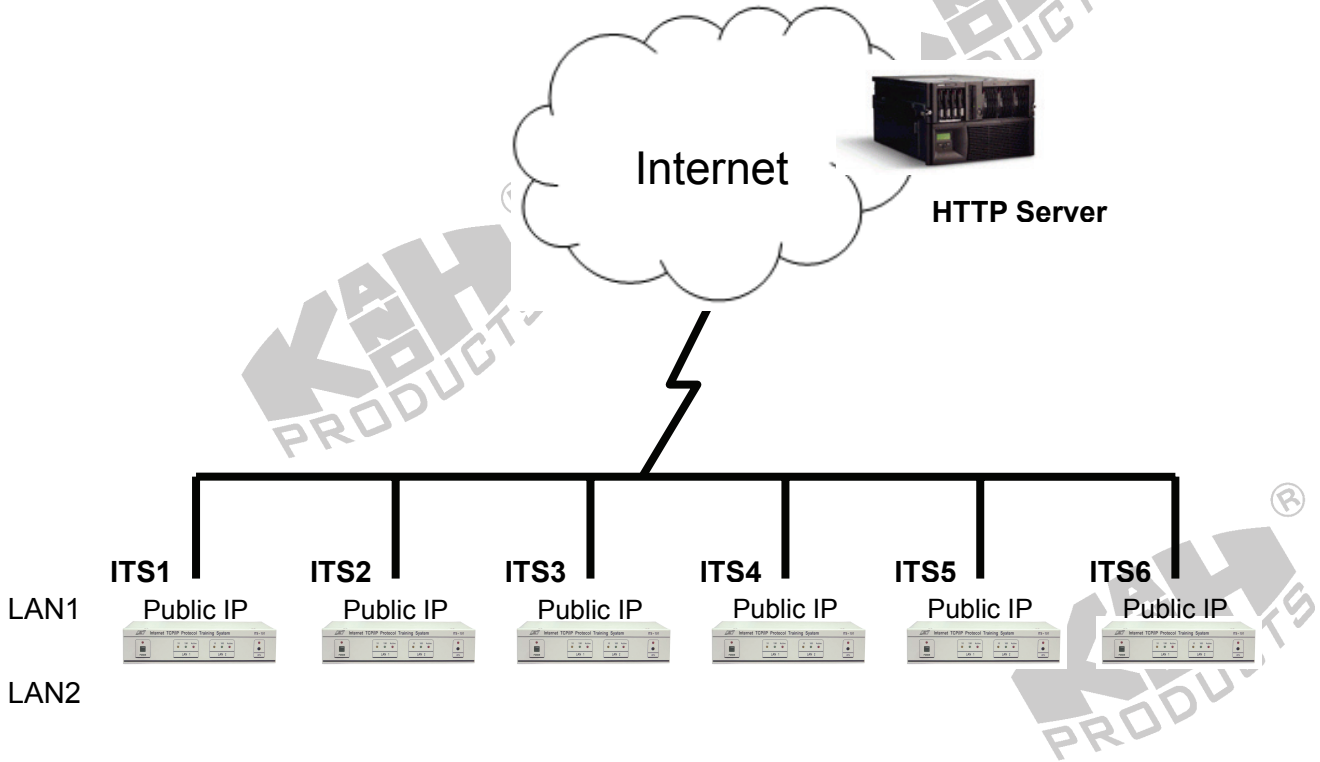
Exp 19. HTTP

OBJECTIVE：To understand the applications of HTTP in TCP.

BRIEF DESCRIPTION：This experiment examines HyperText Transfer Protocol (HTTP) that is used to allow a user to request some hypertext into a computer across an internet. By using **TCPS** GUI tool, students can send HTTP commands to some sound HTTP server to know what HTTP protocol is. Besides, students can also learn how to implement a HTTP client and server.

DURATION：3 hrs

TOPOLOGY



TECHNICAL BACKGROUND

Protocol suite:	TCP/IP
Port:	80: TCP server.

Packet Encapsulation:

MAC header	IP header	TCP header	HTTP header	Data

The HTTP protocol is a request/response protocol. A client sends a request to the server. The server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta-information, and possible entity-body content.

HTTP client establish a TCP connection to port 80 but other ports can be used.

HTTP Procedure

1. The HTTP client initiates a TCP connection to port 80 of the server www.kandh.com.tw.
2. The HTTP client sends a HTTP request message into the TCP connection. The request message either includes the entire URL or simply the path name /some-dir/index.html.
3. The HTTP server receives the request message, retrieves the object /some-dir/index.html from its storage (RAM or disk), encapsulates the object in a HTTP response message, and sends the response message to HTTP client.
4. The HTTP server tells TCP to close the TCP connection. (But TCP doesn't actually terminate the connection, until the client has received the response message in tact.)
5. The HTTP client receives the response message. The TCP connection terminates. The message indicates that the encapsulated object is an HTML file.

HTTP Request

Below we provide a typical HTTP request message:

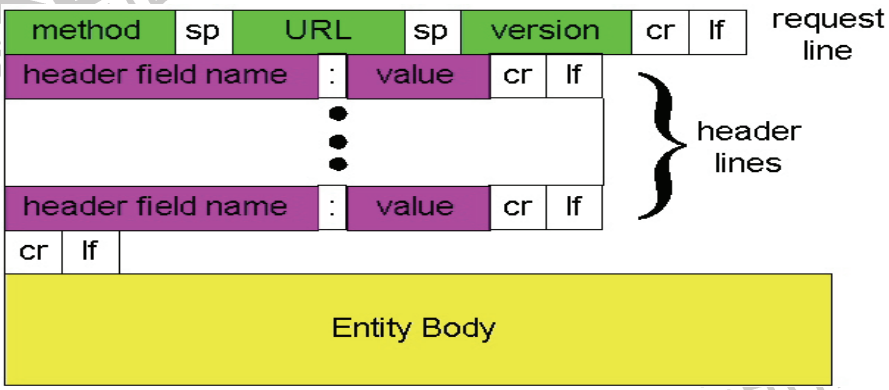
GET /some-dir/index.html HTTP/1.1

Connection: close

User-agent: Mozilla/4.0

Accept: text/html, image/gif, image/jpeg

Accept-language: fr



The method field can take on several different values, including GET, POST, and HEAD. The great majority of HTTP request messages use the GET method. The GET method is used when the browser requests an object, with the object identified in the URL field. In this example, the browser is requesting the object /some-dir/index.html.

HTTP Response Message

This response message could be the response to the example request message just discussed.

HTTP/1.1 200 OK

Connection: close

Date: Thu, 06 Aug 1998 12:00:15 GMT

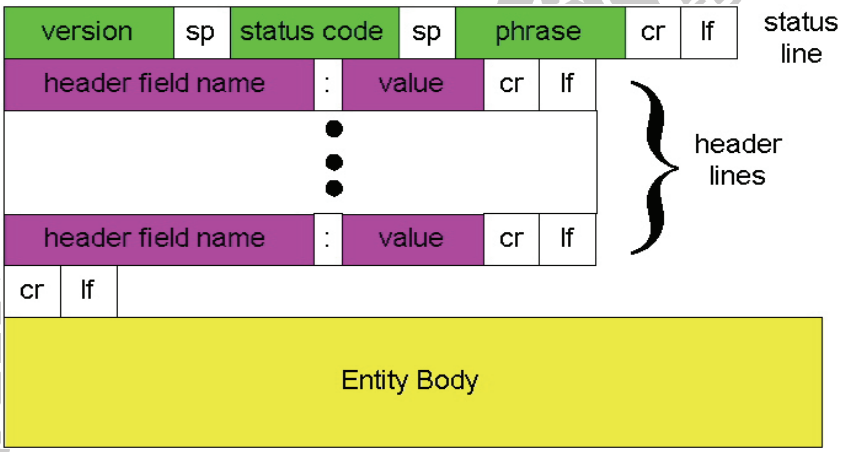
Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT

Content-Length: 6821

Content-Type: text/html

data data data ...



Some example status codes and associated phrases include:

a. 200 OK: Request succeeded and the information is returned in the response.

b. 301 Moved Permanently: Requested object has been permanently moved; new URL is specified in Location: header of the response message. The client software will automatically retrieve the new URL.

c. 400 Bad Request: A generic error code indicating that the request could not be understood by the server.

d. 404 Not Found: The requested document does not exist

e. 505 HTTP Version Not Supported: The request HTTP protocol version is not supported by the server.

PROCEDURE

In this experiment, every ITS needs a real IP address for Internet.

Realizing Network Topology

1. Complete the network connections on HUBOX by referring to Figure 19.1.

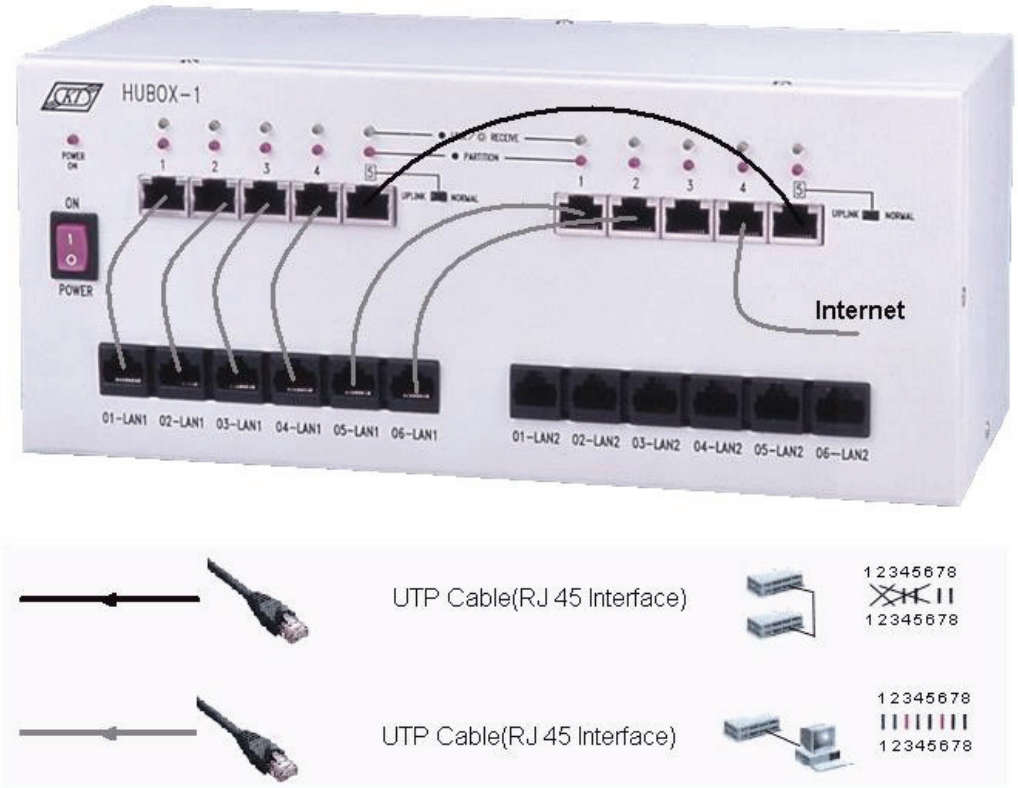


Figure 19.1

Getting Web Page by HTTP

A. Setup

2. Execute **XCLIENT.BAT** to open the KCodes Network Explorer for ITS window.
3. In this training, you needn't to open message browser. Select **Network Configuration** from the Tool menu to open the Network Configuration dialog box.
4. Type <***your Internet IP address***> into IP Address of Interface 1, enter <***your Internet gateway address***> into Routing Table. For example, type "**192.168.1.223**" into IP Address of Interface 1, then enter "**192.168.1.254**" into Gateway and "**0.0.0.0**" into Destination and Mask in the Routing Table. (See Figure 19.2.)

5. Choose **Host**, then click the **Set & Close** button.

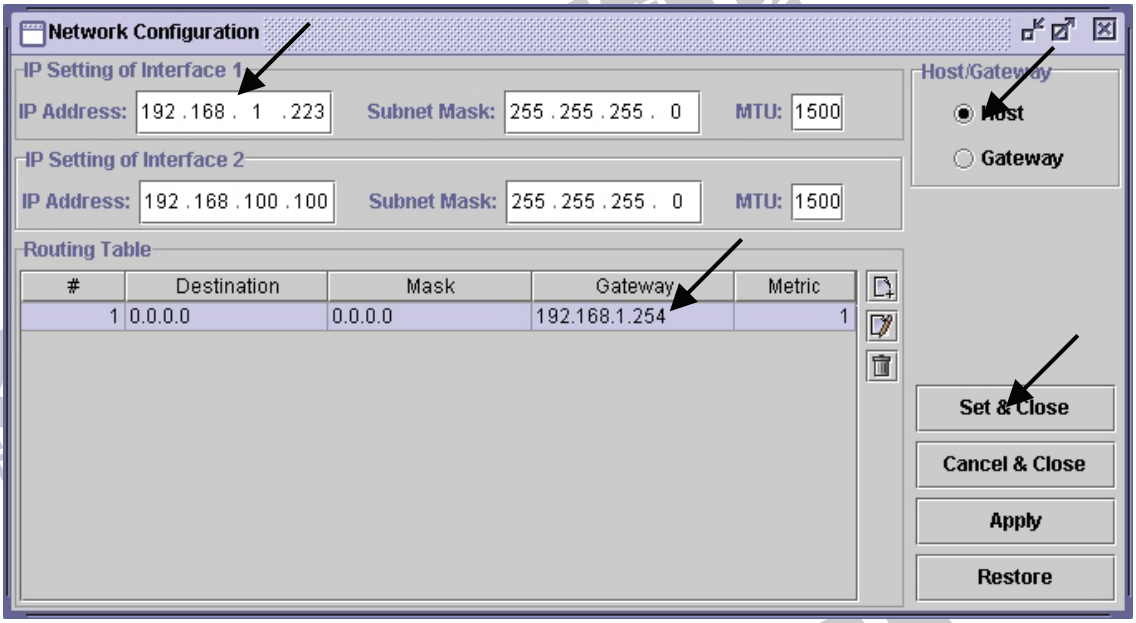


Figure 19.2

B. Get Web Page

6. Open the New TCP Session dialog box by selecting **New TCP Session** from the TCP menu.
7. Select **System Default TCP**. Type <***your SMTP server IP address***> into Destination IP Address, choose **HTTP (80)** from Destination Port. For example, type "**61.218.30.102**" into Destination IP Address.
8. Check **Auto Receive** and click the **Connect** button. Your ITS will be linked to www.kandh.com.tw as shown in Figure 19.3.

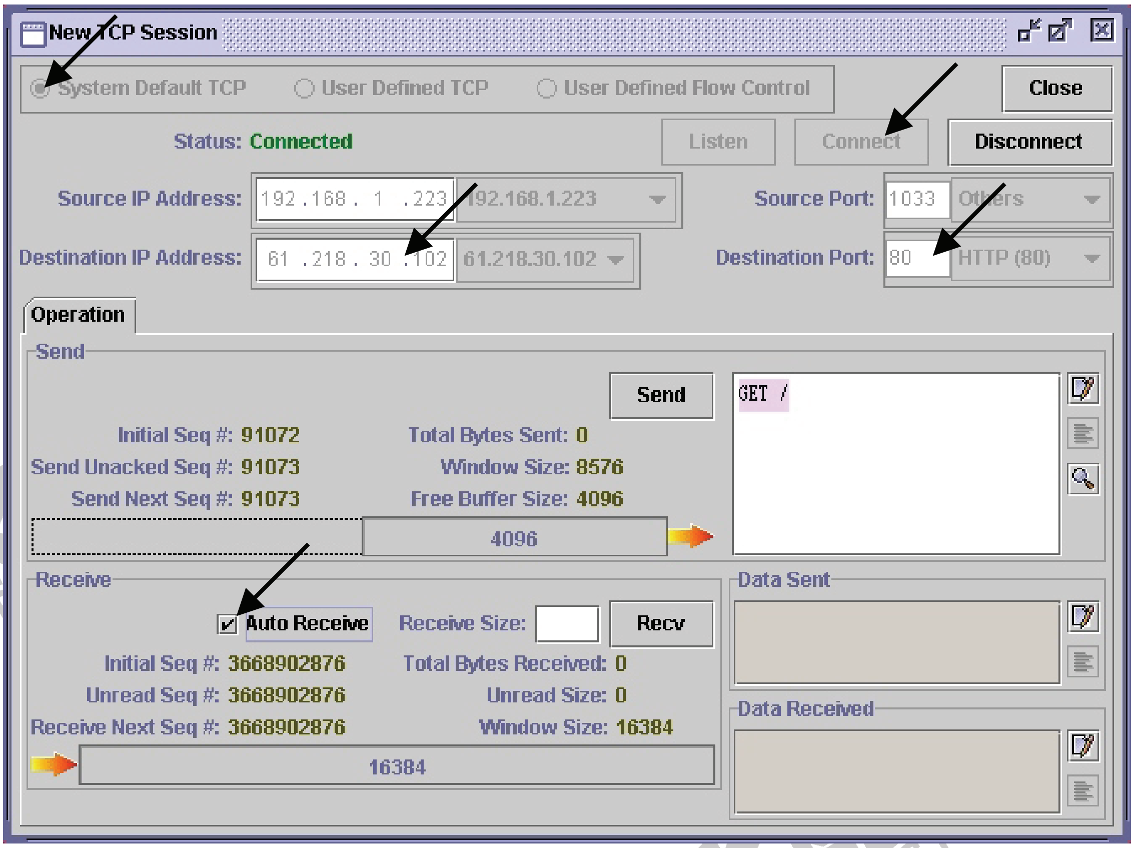


Figure 19.3

9. Type **GET /** ↵ into the edit box, then click the **Send** button. You should receive the web page of www.kandh.com.tw in Data Received mailbox as shown in Figure 19.4.

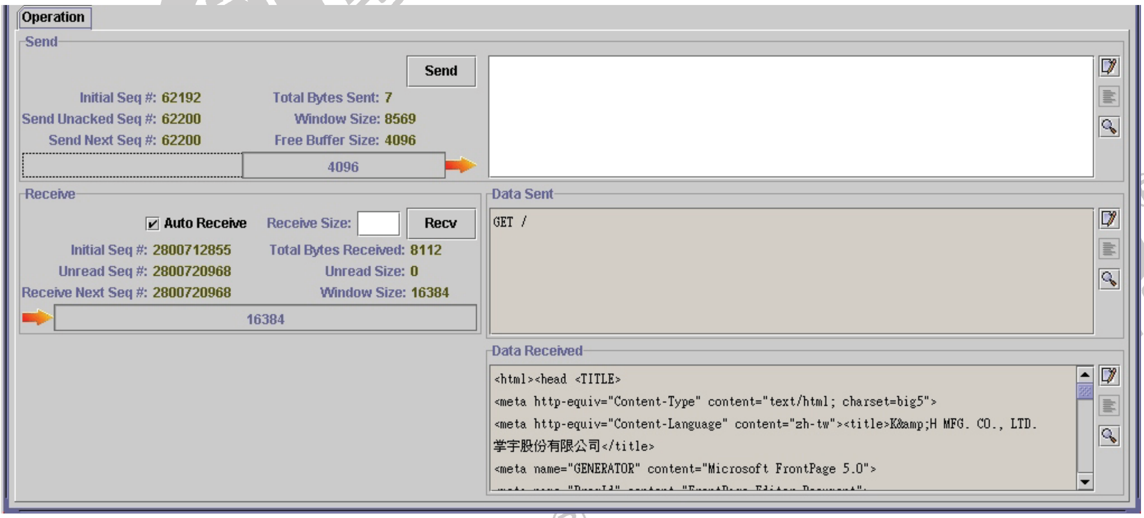


Figure 19.4

DISCUSSIONS

1. Try to type incorrect commands (e.g., gets/, got/, buy\...) into edit box when ITS connects to www.kandh.com.tw. How does web side reply?
2. PC uses the browser (IE or Netscape) to connect www.kandh.com.tw. Then save this web page as a text file (*.txt). Is it the same as ITS Data Received?

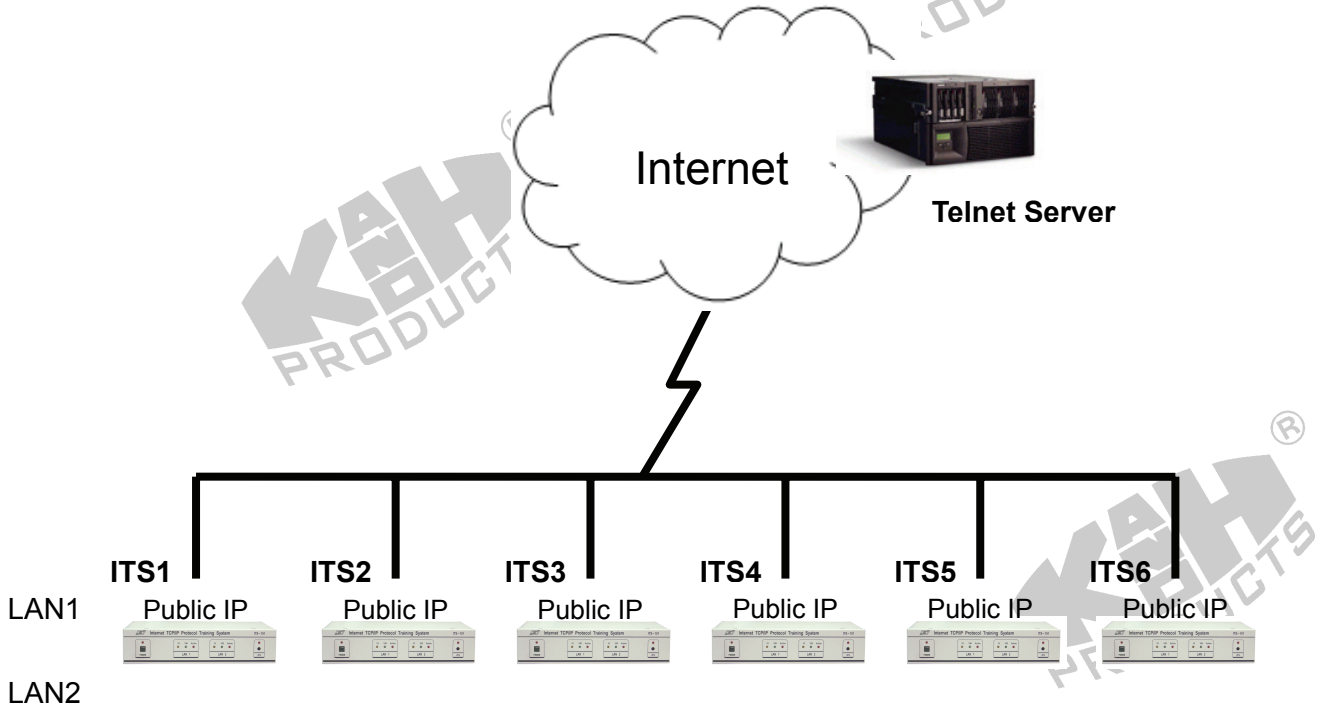
Exp 20. Telnet

OBJECTIVE：To understand what TELNET protocol is and how to implement it.

BRIEF DESCRIPTION： This experiment examines remote terminal protocol called TELNET that is used to allow a user to log into a computer across an internet. By using **TCPS** GUI tool, students can send TELNET control commands to some sound TELNET server to know what TELNET protocol is. Besides, students can also learn how to implement a TELNET client and server.

DURATION：4.5 hrs

TOPOLOGY



TECHNICAL BACKGROUND

Protocol suite:	TCP/IP
Port:	23: TCP server.

Packet Encapsulation:

MAC header	IP header	TCP header	TELNET header	Data

The TCP/IP protocol suite includes a simple remote terminal protocol called *TELNET* that allows a user to log into a computer across an internet. TELNET establishes a TCP connection, and then passes keystrokes from the user's keyboard directly to the remote computer as if they had been typed on a keyboard attached to the remote machine. TELNET also carries output from the remote machine back to the user's screen. The service is called *transparent* because it gives the appearance that the user's keyboard and display attach directly to the remote machine.

The client establishes a TCP connection to the server over which they will communicate. Once the connection has been established, the client accepts keystrokes from the user's keyboard and sends them to the server, while it concurrently accepts characters that the server sends back and displays them on the user's screen. The server must accept a TCP connection from the client, and then relay data between the TCP connection and the local operating system.

The primary goal of the Telnet Protocol is to provide a standard interface for terminal devices and terminal oriented processes on the network. The Telnet Protocol provides a general bidirectional, 8-bit transparent communication channel. The Telnet Protocol is built upon two main ideas: "*Network Virtual Terminal*" and the principle of negotiated options.

The Principle of Negotiated Options

TELNET protocol exchanges “option code sequences” by in-band signaling. TELNET defines a special byte, the Interpret As Command (IAC) with the value 255. When IAC is received, the following byte(s) is interpreted as a TELNET Command. Table lists all the commands defined in RFC 854. Note that in order to send the data byte 255, TELNET must send IAC.

TELNET Commands

Code	Name	Description
240	SE	End of subnegotiation parameters.
241	NOP	No operation.
242	Data Mark	The data stream portion of a Synch. This should always be accompanied by a TCP Urgent notification.
243	Break	NVT character BRK.
244	Interrupt Process	The function IP.
245	Abort output	The function AO.
246	Are You There	The function AYT.
247	Erase character	The function EC.
248	Erase Line	The function EL.
249	Go ahead	The GA signal.
250	SB	Indicates that what follows is subnegotiation of the indicated option.
251	WILL (option code)	Indicates the desire to begin performing, or confirmation that you are now performing, the indicated option.
252	WON'T (option code)	Indicates the refusal to perform, or continue performing, the indicated option.
253	DO (option code)	Indicates the request that the other party perform, or confirmation that you are expecting the other party to perform, the indicated option.
254	DON'T (option code)	Indicates the demand that the other party stop performing, or confirmation that you are no longer expecting the other party to perform, the indicated option.
255	IAC	Data Byte 255.

Telnet options

Code	Option	References
0	TRANSMIT-BINARY, Binary Transmission.	RFC 856
1	ECHO, Echo.	RFC 857
2	Reconnection.	
3	SUPPRESS-GO-AHEAD, Suppress Go Ahead.	RFC 858
4	Approx Message Size Negotiation.	
5	STATUS.	RFC 859
6	TIMING-MARK, Timing Mark	RFC 860
7	RCTE, Remote Controlled Trans and Echo.	RFC 563, RFC 726
8	Output Line Width.	
9	Output Page Size.	
10	NAOCRD, Negotiate About Output Carriage-Return Disposition.	RFC 652
11	NAOHTS, Negotiate About Output Horizontal Tabstops.	RFC 653
12	NAOHTD, Negotiate About Output Horizontal Tab Disposition.	RFC 654
13	NAOFFD, Negotiate About Output Formfeed Disposition.	RFC 655
14	NAOVTS, Negotiate About Vertical Tabstops.	RFC 656
15	NAOVTD, Negotiate About Output Vertcial Tab Disposition.	RFC 657
16	NAOLFD, Negotiate About Output Linefeed Disposition.	RFC 658
17	Extended ASCII.	RFC 698
18	LOGOUT, Logout.	RFC 727
19	BM, Byte Macro.	RFC 735
20	Data Entry Terminal.	RFC 732, RFC 1043
21	SUPDUP.	RFC 734, RFC 736
22	SUPDUP-OUTPUT, SUPDUP Output.	RFC 749
23	SEND-LOCATION, Send Location.	RFC 779
24	TERMINAL-TYPE, Terminal Type.	RFC 1091
25	END-OF-RECORD, End of Record.	RFC 885
26	TUID, TACACS User Identification.	RFC 927
27	OUTMRK, Output Marking.	RFC 933
28	TTYLOC, Terminal Location Number.	RFC 946
29	Telnet 3270 Regime.	RFC 1041
30	X.3 PAD.	RFC 1053
31	NAWS, Negotiate About Window Size.	RFC 1073
32	Terminal Speed.	RFC 1079
33	Remote Flow Control.	RFC 1372
34	Line mode.	RFC 1184
35	X Display Location.	RFC 1096

36	Environment Option.	RFC 1408
37	AUTHENTICATION, Authentication Option.	RFC 1416, RFC 2941, RFC 2942, RFC 2943, RFC 2951
38	Encryption Option.	RFC 2946
39	New Environment Option.	RFC 1572
40	TN3270E.	RFC 2355
41	XAUTH.	
42	CHARSET.	RFC 2066
43	RSP, Telnet Remote Serial Port.	
44	Com Port Control Option	RFC 2217
45	Telnet Suppress Local Echo	
46	Telnet Start TLS	
47	KERMIT	RFC 2840
48	SEND-URL	
49	FORWARD_X	
50 - 137		
138	TELOPT PRAGMA LOGON	
139	TELOPT SSPI LOGON	
140	TELOPT PRAGMA HEARTBEAT	
141 - 254		
255	EXOPL, Extended-Options-List.	RFC 861

PROCEDURE

In this experiment, every ITS needs a real IP address for Internet.

Realizing Network Topology

1. Complete the network connections on HUBOX by referring to Figure 20.1.

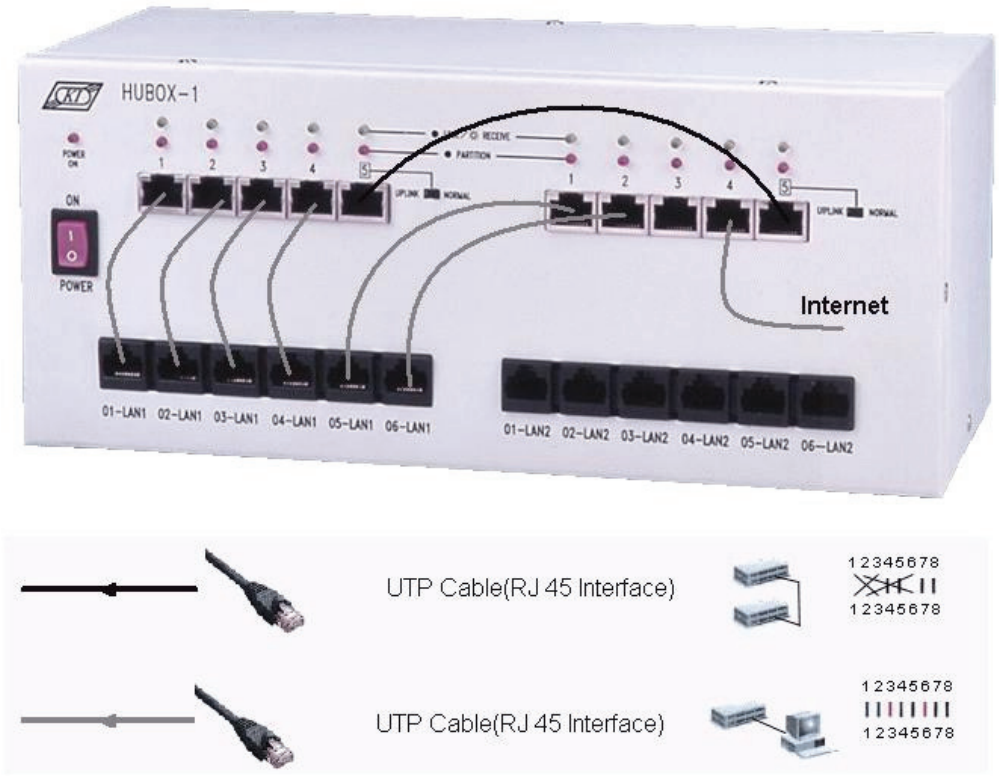


Figure 20.1

TELNET Login

A. Setup

2. Execute **XCLIENT.BAT** to open the KCodes Network Explorer for ITS window.
3. Open the Network Message Browser window by selecting **New Memorized Message Browser** from the Listen menu.
4. Choose **Option** from Network Message Browser to open the Set Message Range dialog box.
5. Click the **Add new rule** button. You need to set two rules for message browser. First type "**23**" into Remote Port, then click the **Apply** button. Secondly type "**23**" into Local

Port, then click the **Apply** button again. (See Figure 20.2.)

6. Finally click the **Set & Close** button.

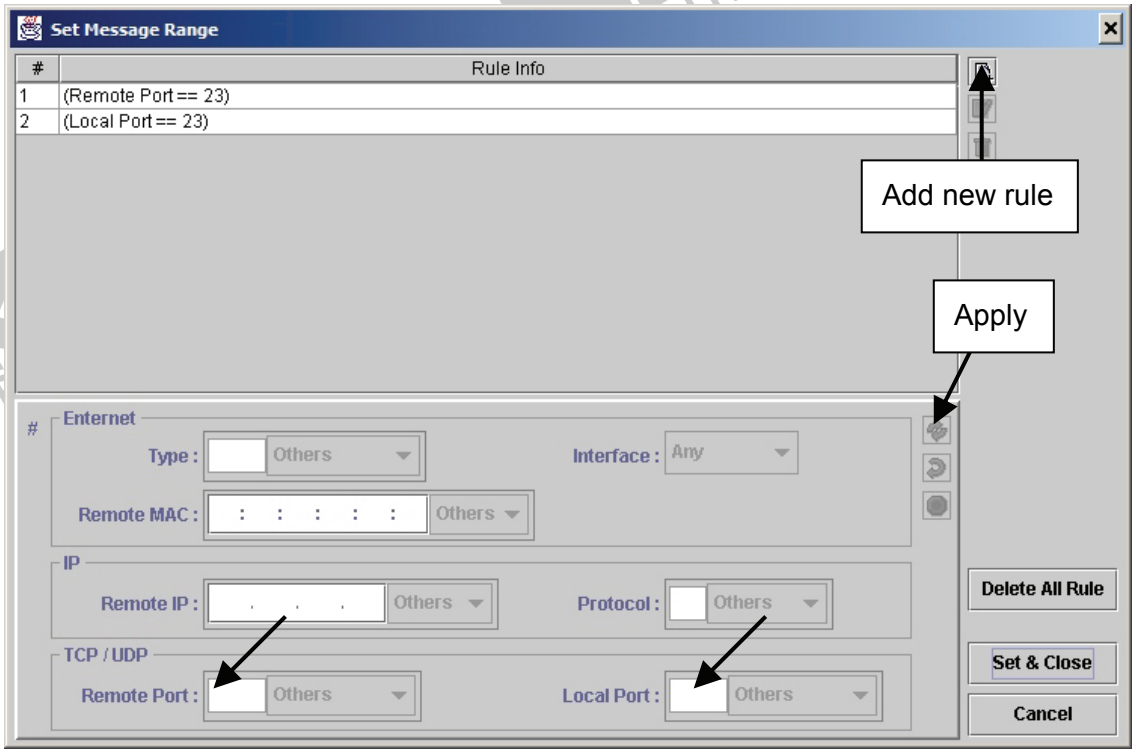


Figure 20.2

7. Select **Network Configuration** from the Tool menu to open the Network Configuration dialog box.

8. Type <***your Internet IP address***> into IP Address of Interface 1, enter <***your Internet gateway address***> into Routing Table. For example, type "**192.168.253.1**" into IP Address of Interface 1, and enter "**192.168.253.254**" into Gateway and "**0.0.0.0**" into Destination and Mask in the Routing Table. (See Figure 20.3.)

9. Choose **Host**, then click the **Set & Close** button.

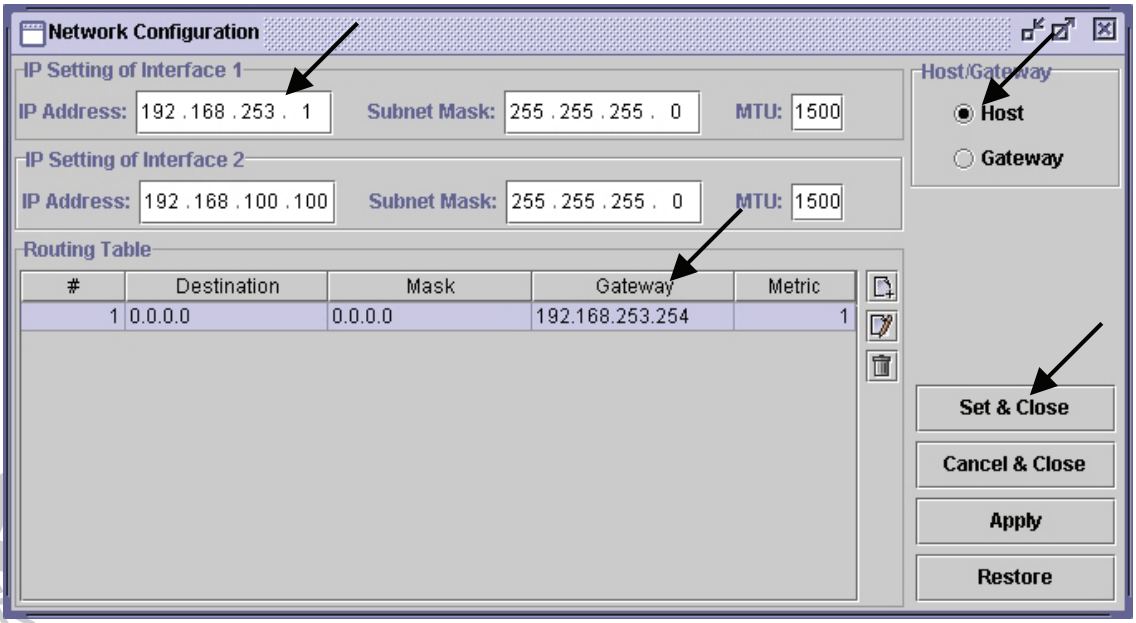


Figure 20.3

B. Login TELNET

10. Open the New TCP Session dialog box by selecting **New TCP Session** from the TCP menu.

11. Select **System Default TCP**. Type <***your TELNET server IP address***> into Destination IP Address, choose **TELNET (23)** from Destination Port. For example, type "**203.149.174.99**" into Destination IP Address as shown in Figure 20.4.

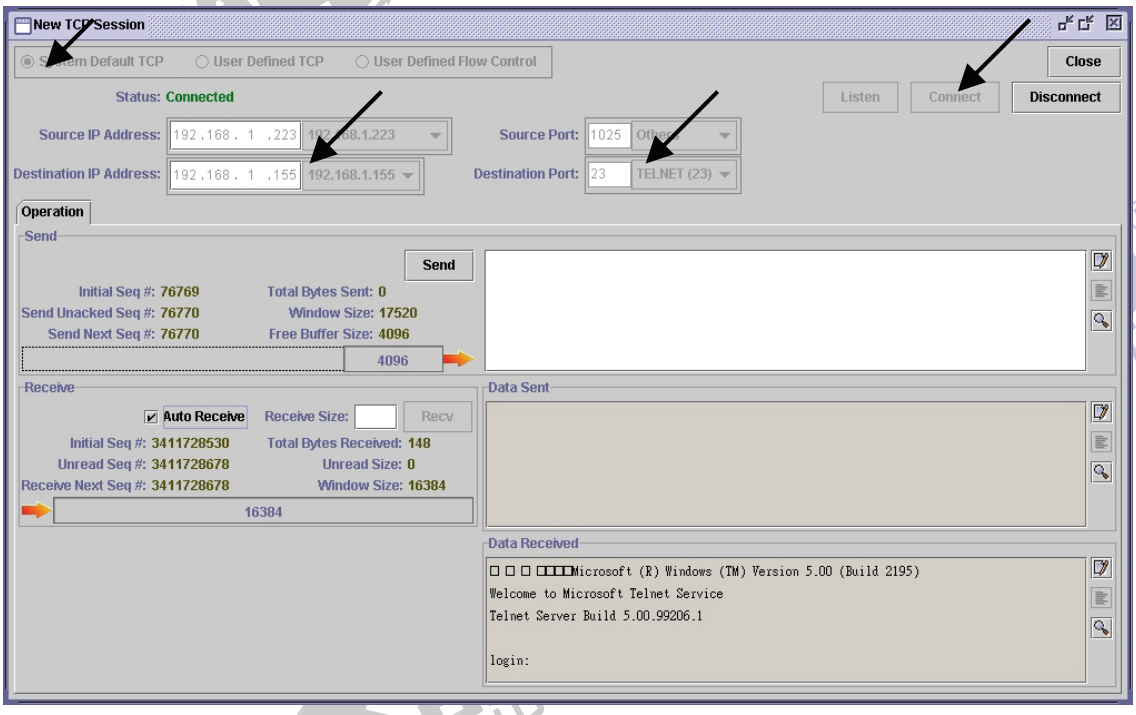
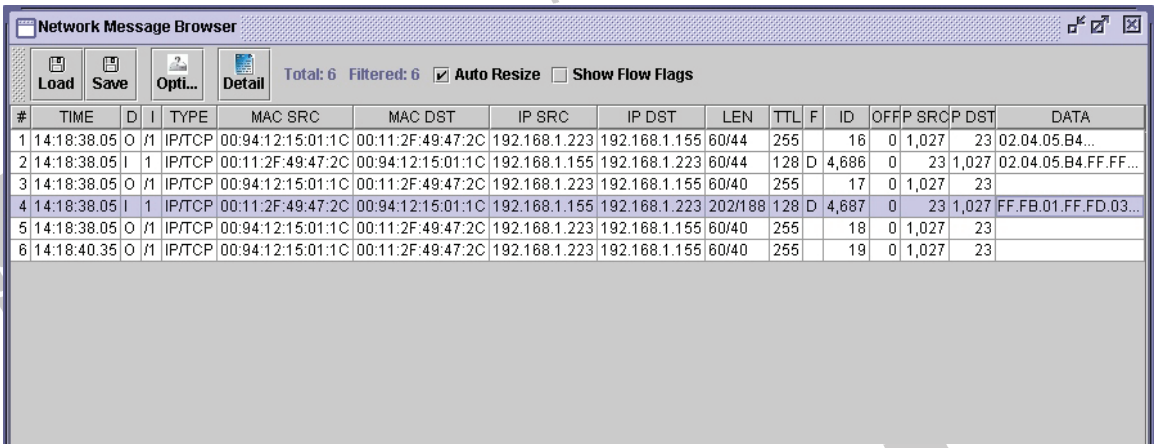


Figure 20.4

12. Click the **Connect** button. You can see the procedure of TELNET setup connection from message browser as shown in Figures 20.5 and 20.6.



#	TIME	D	I	TYPE	MAC SRC	MAC DST	IP SRC	IP DST	LEN	TTL	F	ID	OFF	P SRC	P DST	DATA
1	14:18:38.05	O	/1	IP/TCP	00:94:12:15:01:1C	00:11:2F:49:47:2C	192.168.1.223	192.168.1.155	60/44	255		16	0	1,027	23	02.04.05.B4...
2	14:18:38.05	I	1	IP/TCP	00:11:2F:49:47:2C	00:94:12:15:01:1C	192.168.1.155	192.168.1.223	60/44	128	D	4,686	0	23	1,027	02.04.05.B4.FF.FF...
3	14:18:38.05	O	/1	IP/TCP	00:94:12:15:01:1C	00:11:2F:49:47:2C	192.168.1.223	192.168.1.155	60/40	255		17	0	1,027	23	
4	14:18:38.05	I	1	IP/TCP	00:11:2F:49:47:2C	00:94:12:15:01:1C	192.168.1.155	192.168.1.223	202/188	128	D	4,687	0	23	1,027	FF.FB.01.FF.FD.03...
5	14:18:38.05	O	/1	IP/TCP	00:94:12:15:01:1C	00:11:2F:49:47:2C	192.168.1.223	192.168.1.155	60/40	255		18	0	1,027	23	
6	14:18:40.35	O	/1	IP/TCP	00:94:12:15:01:1C	00:11:2F:49:47:2C	192.168.1.223	192.168.1.155	60/40	255		19	0	1,027	23	

Figure 20.5

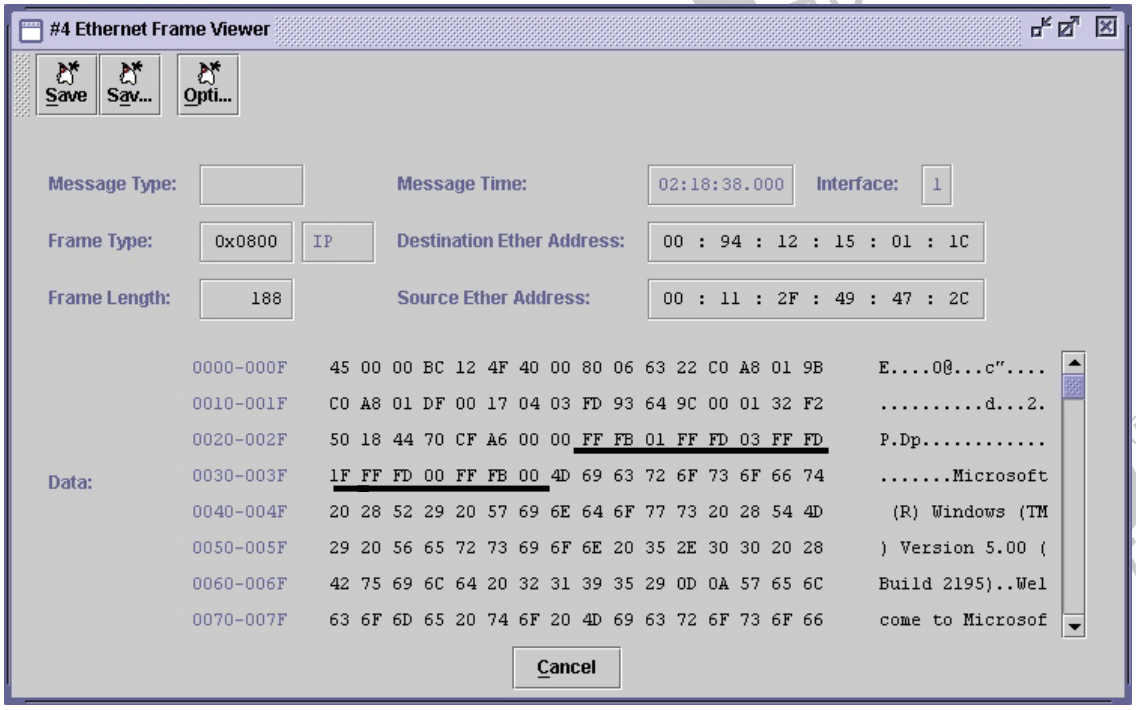


Figure 20.6

DISCUSSION

1. With different systems, the client exchanges “option code sequences” with the server as follows:

Sequence	Direction (send)	Options in Text Mode	Options in Binary Mode
1	Server	DO TERMINAL TYPE	FF FD 18
2	Server	DO TERMINAL SPEED	FF FD 20
3	Server	DO X DISPLAY LOCATION	FF FD 23
4	Server	DO NEW ENVIRINMENT OPTION.	FF FD 27
5	Client	WILL TERMINAL TYPE	FF FB 18
6	Client	WONT TERMINAL SPEED	FF FC 20
7	Client	WONT X DISPLAY LOCATION	FF FC 23
8	Client	WILL NEW ENVIRINMENT OPTION.	FF FB 27
9	Client	WILL NAWS (Negotiate About Window Size)	FF FB 1F
10	Server	DO NAWS (Negotiate About Window Size)	FF FD 1F
11	Client	SB NAWS 80 x 25 SE	FF FA 1F 00 50 00 19 FF F0
12	Server	IAC SB NEW ENVIRINMENT SEND	FF FA 27 01 FF F0
9	Server	IAC SB TERMINAL TYPE SEND	FF FA 18 01 FF F0
10	Client	IAC SB NEW ENVIRINMENT IS	FF FA 27 00 FF F0
11	Client	IAC SB TERMINAL TYPE IS “ANSI”	FF FA 18 00 41 4E 53 49 FF F0
12	Server	WILL SUPPRESS GO AHEAD	FF FB 03
13	Server	DO ECHO	FF FD 01
14	Server	WILL STATUS	FF FB 05
15	Server	DO LFLOW	FF FD 21
16	Client	DO SUPPRESS GO AHEAD	FF FD 03
17	Client	WILL ECHO	FF FB 01
18	Client	DONT STATUS	FF FE 05
19	Client	WONT LFLOW	FF FC 21
20	Server	DONT ECHO	FF FE 01
21	Server	WILL ECHO	FF FB 01

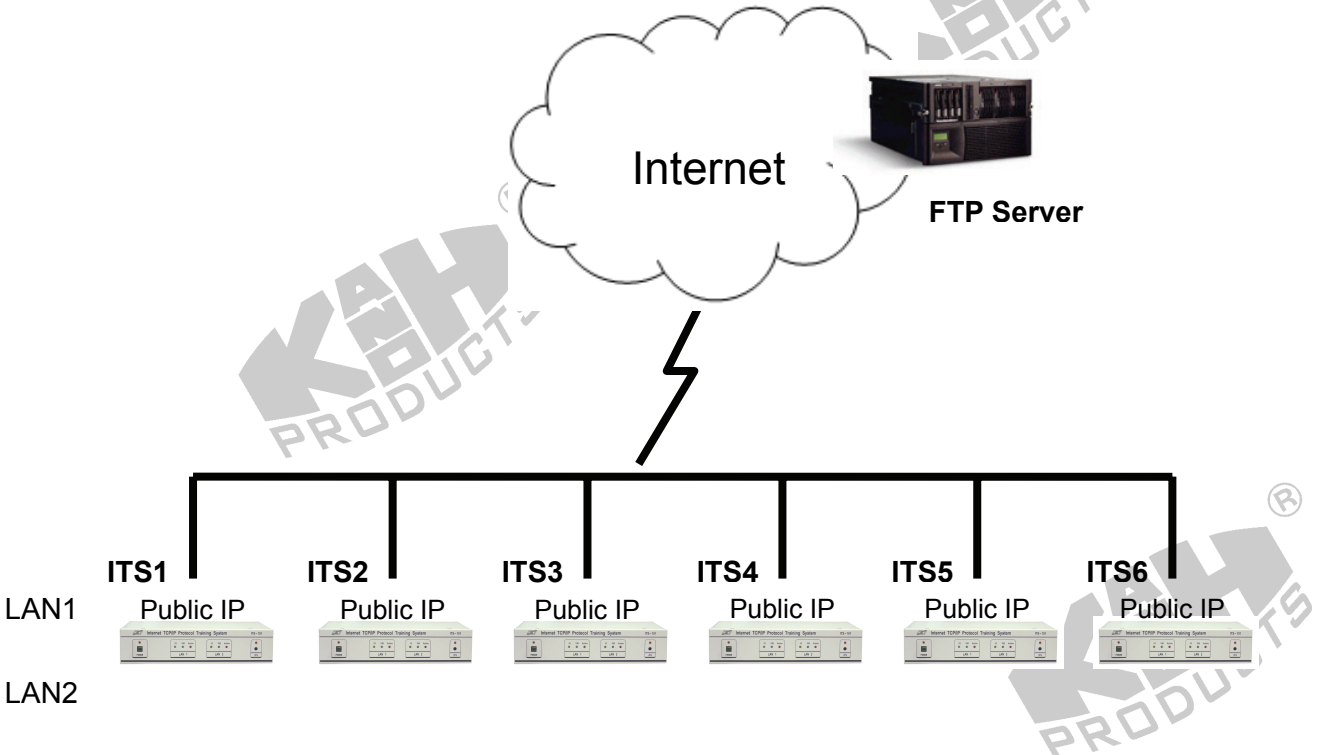
Exp 21. FTP

OBJECTIVE：To understand what FTP protocol is and how to implement it.

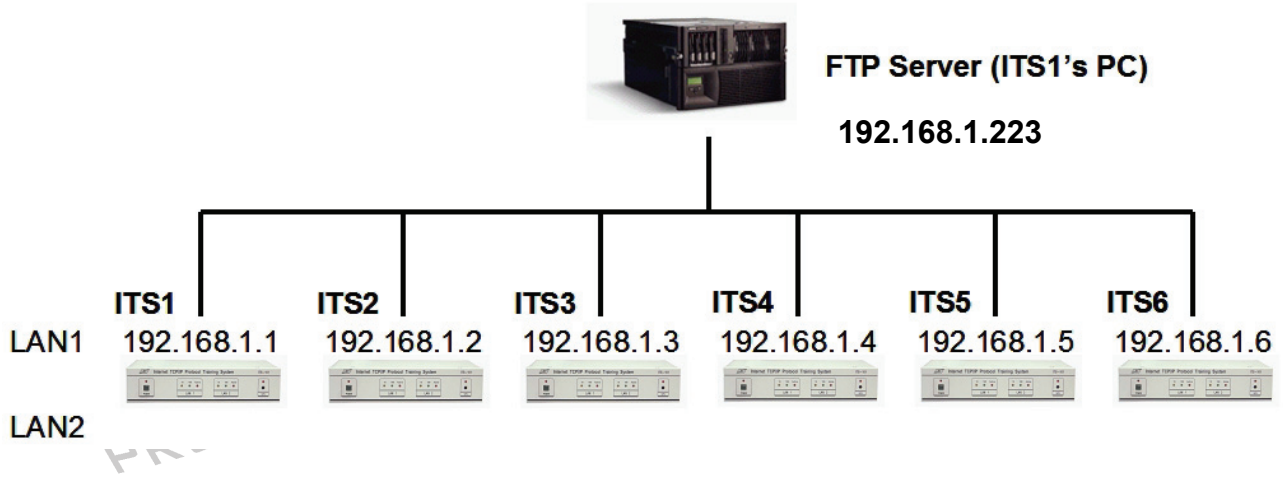
BRIEF DESCRIPTION：This experiment examines the file transfer protocols (FTP) that is used to transfer files between computers and is defined in RFC 959. By using **TCPS** GUI tool, students can send FTP control commands to some sound FTP server to know what FTP protocol is. Besides, students can also learn how to implement a FTP client and server.

DURATION: 4.5 hrs

TOPOLOGY A (Internet FTP Server)



TOPOLOGY B (Local PC FTP server)



TECHNICAL BACKGROUND

Protocol suite:	TCP/IP
Port:	21: FTP Server

Packet Encapsulation:

MAC header	IP header	TCP header	FTP header	Data

RFC959

This experiment examines the file transfer protocols (FTP) that is used to transfer files between computers. FTP uses TCP as a transport protocol to provide reliable end-to-end connections. Two connections are used: the first is *control connection* for login and follows the TELNET protocol and the second is *data connection* for managing the data transfer. As it is necessary to log into the remote host, the user must have a user name and a password to access files and directories.

FTP application is built with a protocol interpreter (PI), a data transfer process (DTP), and a user interface (see Figure 21.1). The user interface communicates with the protocol interpreter, which handles the control connection. Furthermore, the protocol interpreter has to initiate the data

connection. During the file transfer, the data management is performed by DTP. After a user's request is completed, the server's PI has to close the control connection.

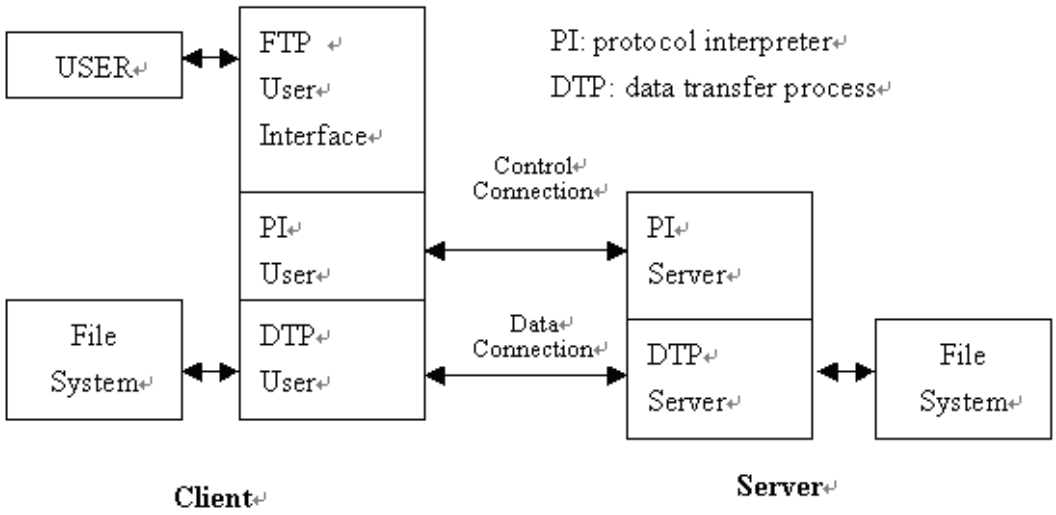


Figure 21.1

PROCEDURE

Realizing Network Topology

1. Complete the network connections on HUBOX by referring to Figure 21.2.

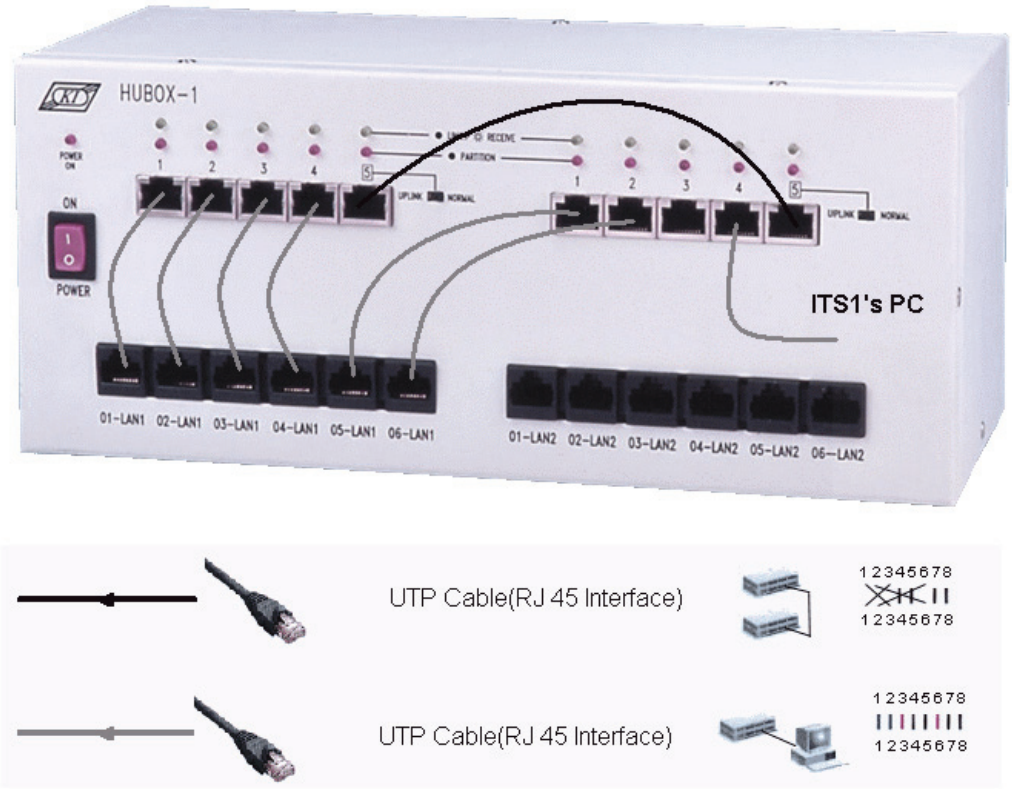


Figure 21.2

FTP Login

A. Setup

Refer to Topology B. In this experiment, the PC of ITS1 is the FTP server. ITS1 thru ITS6 are FTP clients.

PC of ITS1

2. Make sure your system already set up FTP server. (Refer to Appendix B, TYPsoftFTP server)
3. Click the **Start** button on the taskbar to open the start menu. Open the Settings menu and select **Network Connections**. The Network Connections window opens as shown in Figure 21.3.

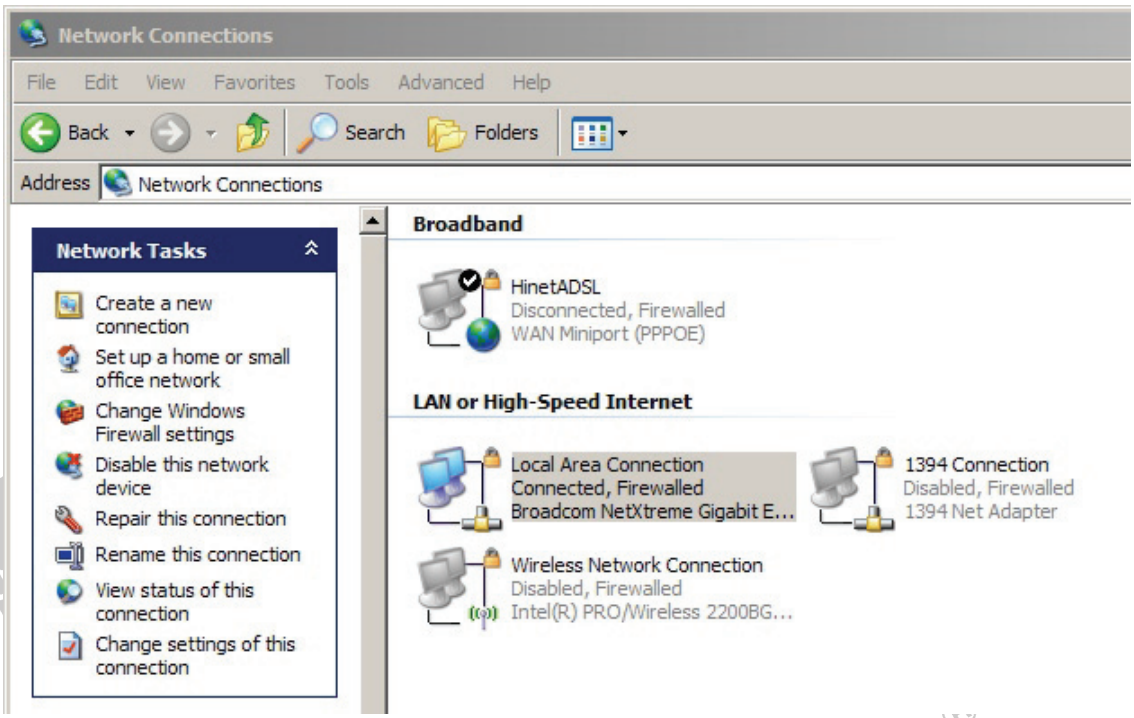


Figure 21.3

4. In the Network Connections window, double-click the **Local Area Connection** icon to open the Local Area Connection Properties window as shown in Figure 21.4.

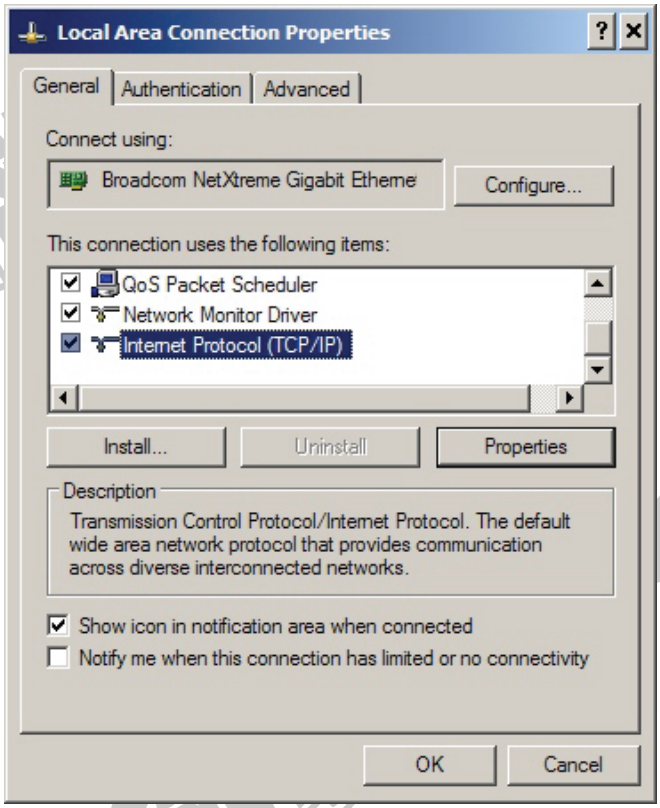


Figure 21.4

5. Select **Internet Protocol (TCP/IP)** and click the **Properties** button. The Internet Protocol (TCP/IP) Properties window opens. Choose the **Use the following IP address** option, set computer network interface as subnet 192.168.1.0 /24. For example, type "**192.168.1.223**" into IP address (See Figure 21.5), and click the **OK** button. Now we already set up the address of FTP server.

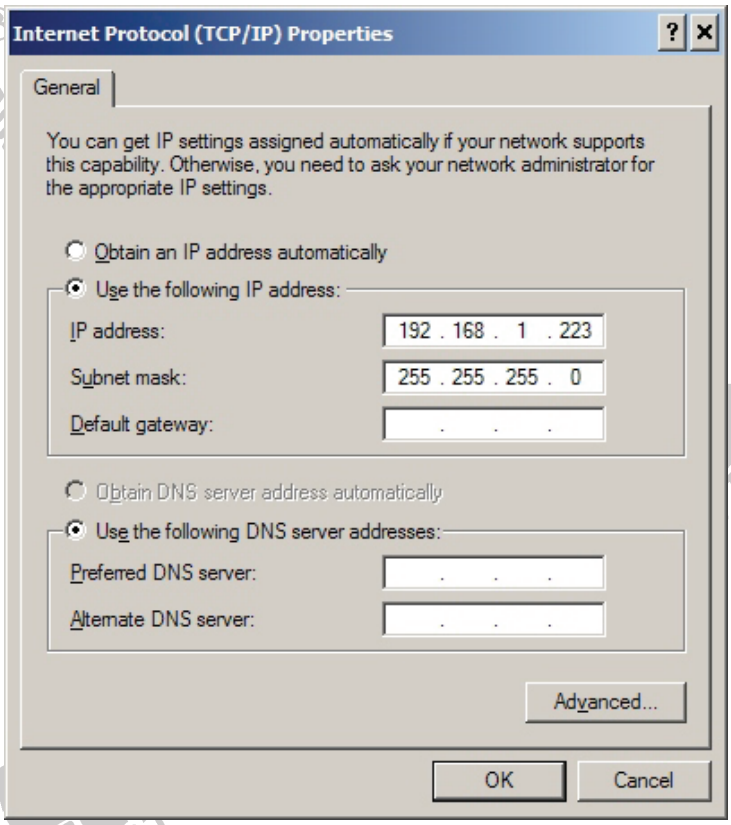


Figure 21.5

ITS1 thru ITS6

6. Choose **Network Configuration** from the Tool menu to open the Network Configuration dialog box.
7. Follow the Topology B to set up the IP address. (See Figure 21.6.)
8. Set **Host**, and click the **Set & Close** button.

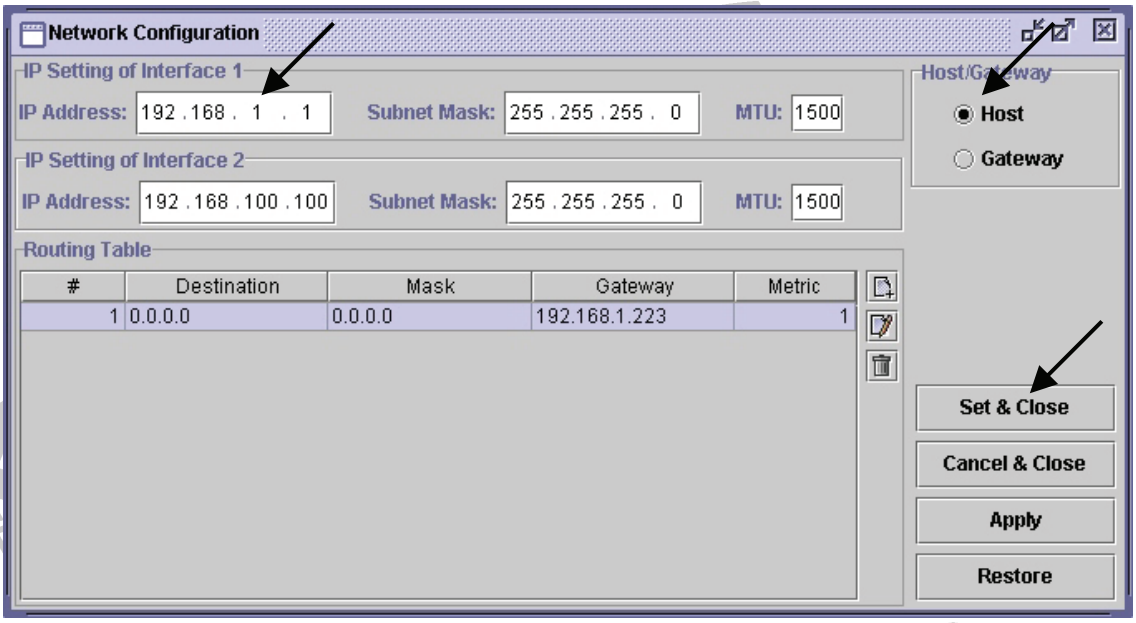


Figure 21.6

B. Login FTP Station

9. Open the New TCP Session dialog box by selecting **New TCP Session** from the TCP menu.
10. Select **System Default TCP**, type <***your IP address***> into Source IP Address, and enter <***your default listen port***> into Source Port. Then click the **Listen** button. For example, type "**192.168.1.1**" into Source IP Address, type "**1029**" into Source Port. Your ITS will open port 1029 that is in listening. (See Figure 21.7.)

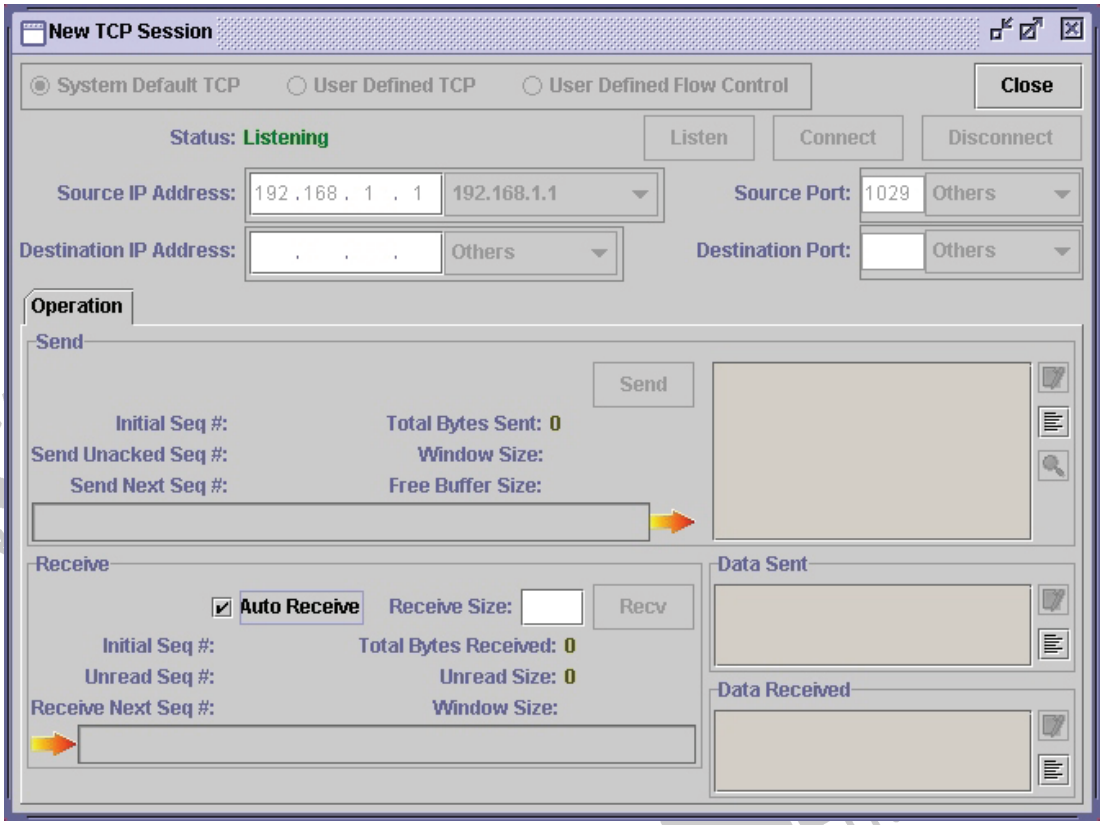


Figure 21.7

11. Open another New TCP Session dialog box.
12. Select **System Default TCP**, type "**192.168.1.223**" into Destination IP Address, choose **FTP (21)** from Destination Port, and click the **Connect** button. You will see the welcome message from FTP server as shown in the Data Received mailbox (see Figure 21.8).

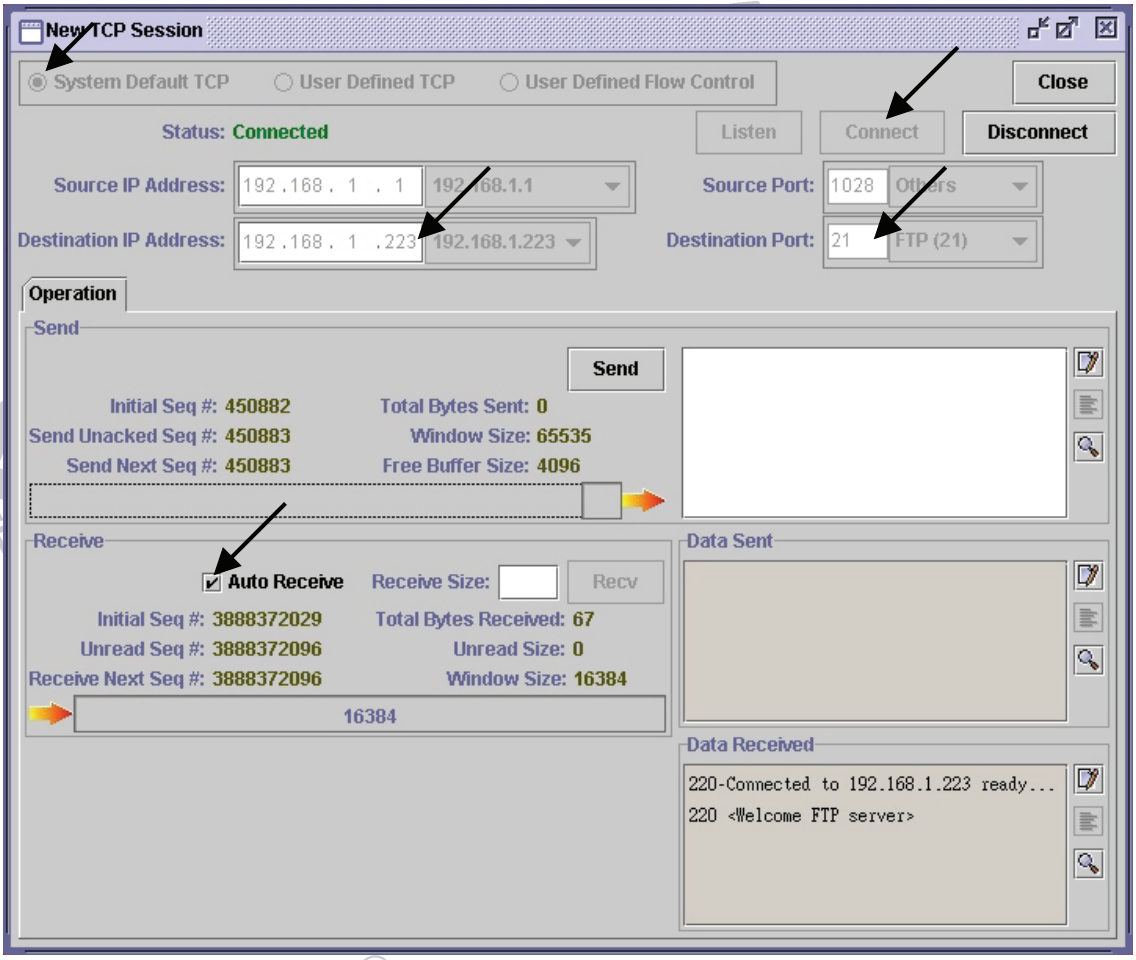


Figure 21.8

13. Referring to Figure 21.9, type "**user <*your account name*>** ↵ " into edit box and click the **Send** button. For example, type "**user kandh**" into edit box.

14. Type "**pass <*your account password*>** ↵ " into edit box, then click the **Send** button again. For example, type "**pass test**" into edit box. Then you can see that your ITS already logins FTP server as shown in Figure 21.10.

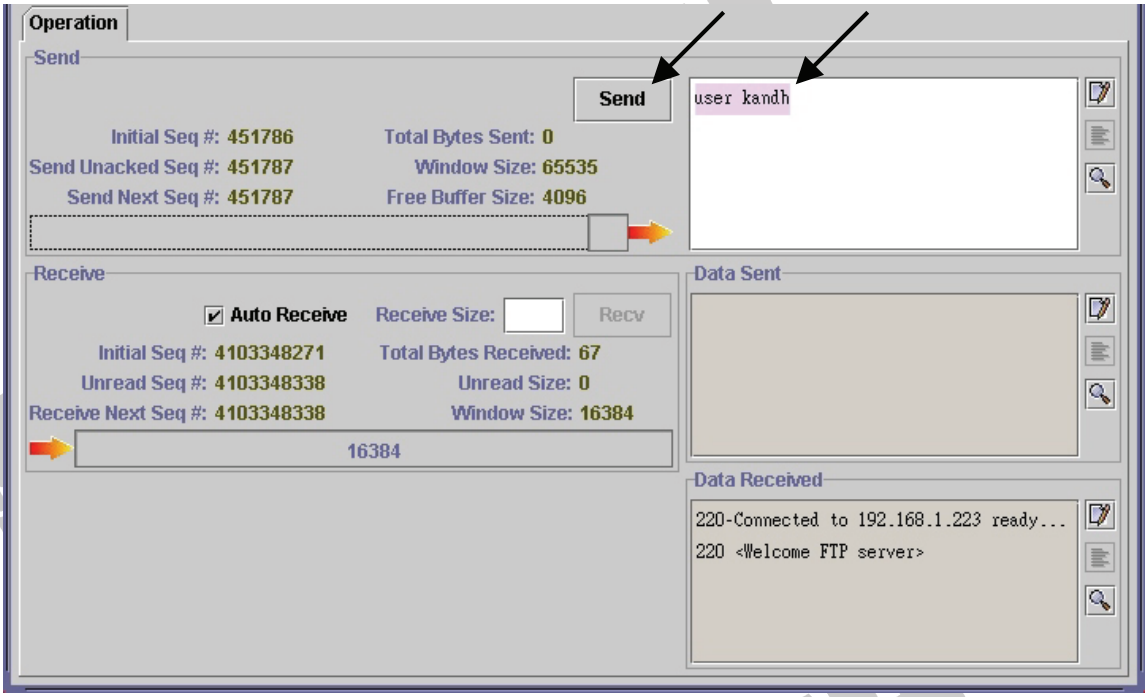


Figure 21.9

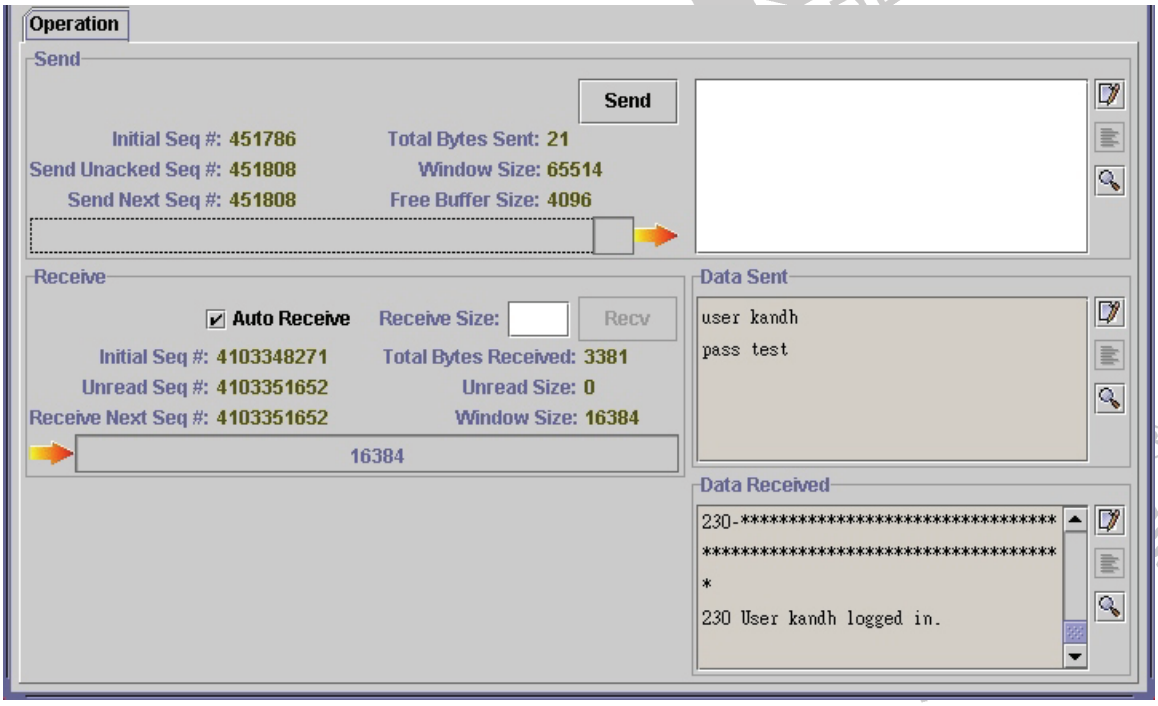


Figure 21.10

15. Type "**port 192,168,1,1,0,1029** ↵" into edit box and click the **Send** button. Type "**list** ↵" into edit box and click the **Send** button again. ITS will send a request to FTP server.

16. From the listen TCP session, you can see that ITS receives data from FTP server as shown in Figure 21.11.

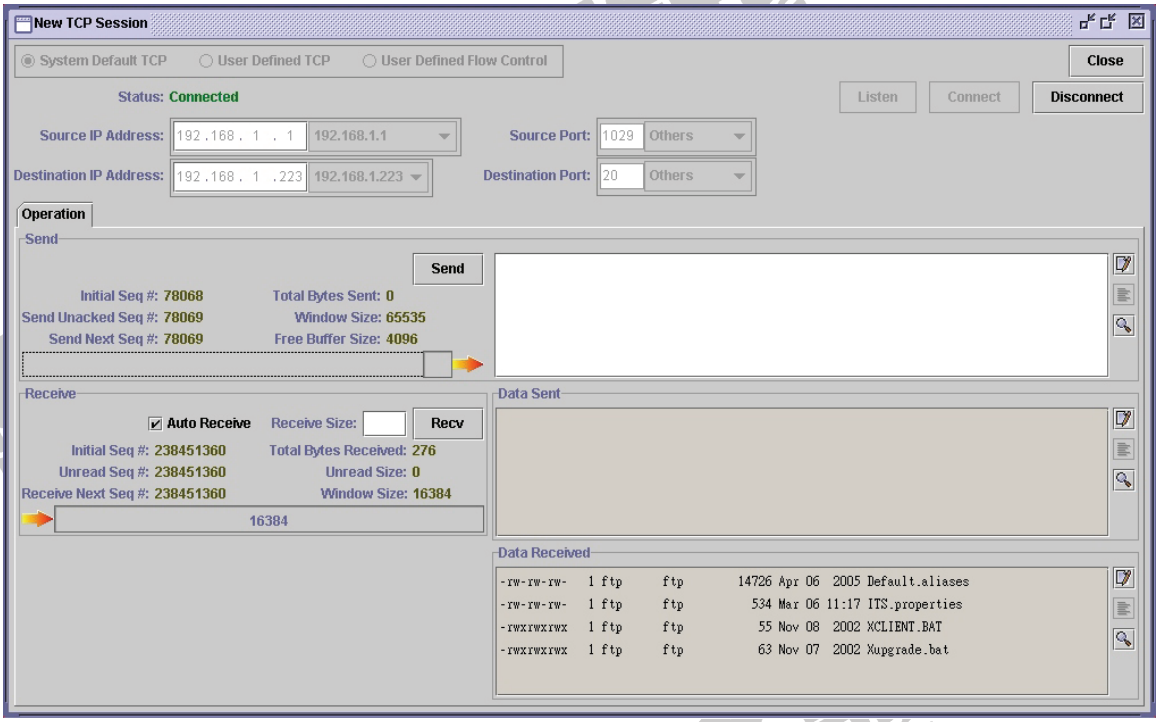


Figure 21.11

DISCUSSION

1. Refer to Topology A. Try to open a TCP active connection by setting destination port to 21 and destination IP address as a public FTP sever of Internet.