



Computer Systems Engineering Technology CST 417 – Embedded Networking

Lab 3
Fall 2014
Instructor: Troy Scevers

Name _____
Due Date: 10/28/2014

Instructions

In this lab we will be modifying the application from lab2. We will re-enable sections of the TCP/IP stack. We will remove the code for checking buttons from the main program. Answer the questions for each section and demo the sections that say demo to the lab instructor.

Procedure

Part 1

Open `main_demo.h` and find the following section and make it look as below.

```
// enable the demo-applications that you want to run
#define APP_USE_GENERIC_TCP_CLIENT_DEMO
#define APP_USE_GENERIC_TCP_SERVER_DEMO
// #define APP_USE_GENERIC_SSL_CLIENT_DEMO
// #define APP_USE_SMTP_CLIENT_DEMO
// #define APP_USE_PING_DEMO
// #define APP_USE_SNMP_TRAP_DEMO
// #define APP_USE_SNMP_V2_TRAP_DEMO
// #define APP_USE_BERKELEY_API_DEMO
```

Next open `tcip_config.h` and make it look like below.

```
// TCPIP Stack Module Selection
// Uncomment or comment the following lines to enable or
// disabled the following high-level application modules.

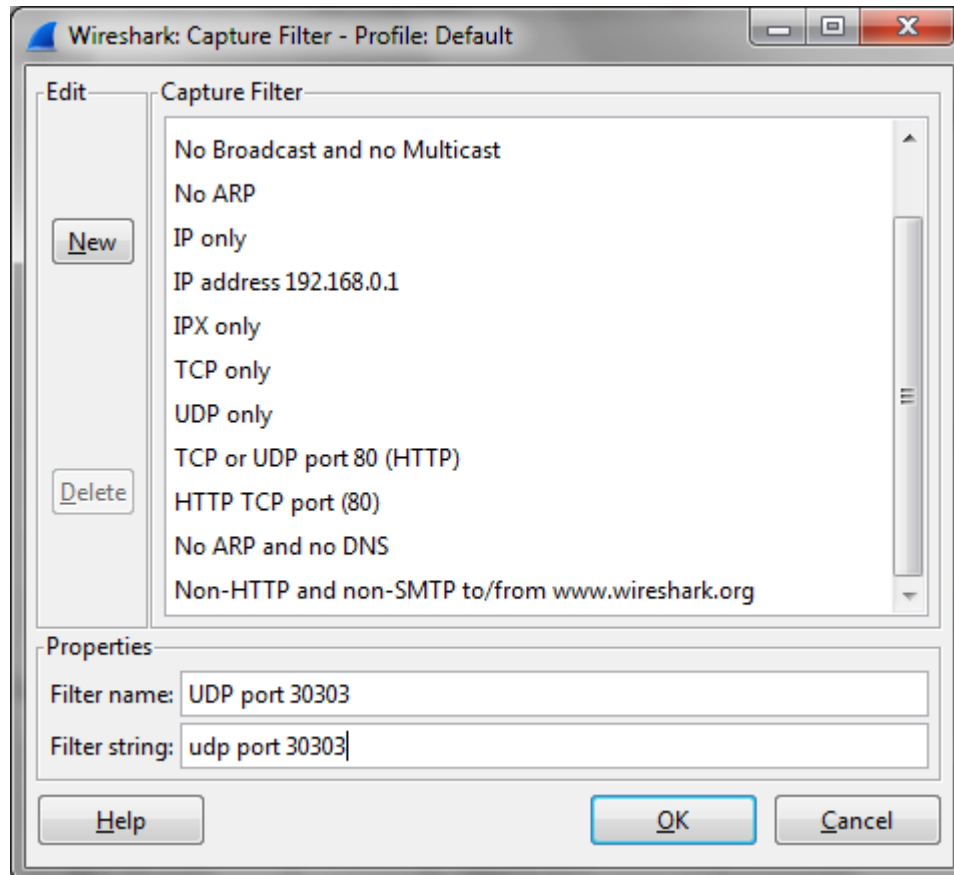
#define TCPIP_STACK_USE_IPV4
#define TCPIP_STACK_USE_ICMP_SERVER
// #define TCPIP_STACK_USE_HTTP2_SERVER
// #define TCPIP_STACK_USE_SSL_SERVER
// #define TCPIP_STACK_USE_SSL_CLIENT
#define TCPIP_STACK_USE_DHCP_CLIENT
// #define TCPIP_STACK_USE_SMTP_CLIENT
// #define TCPIP_STACK_USE_TELNET_SERVER
#define TCPIP_STACK_USE_ANNOUNCE
#define TCPIP_STACK_USE_DNS
#define TCPIP_STACK_USE_NBNS
```

```

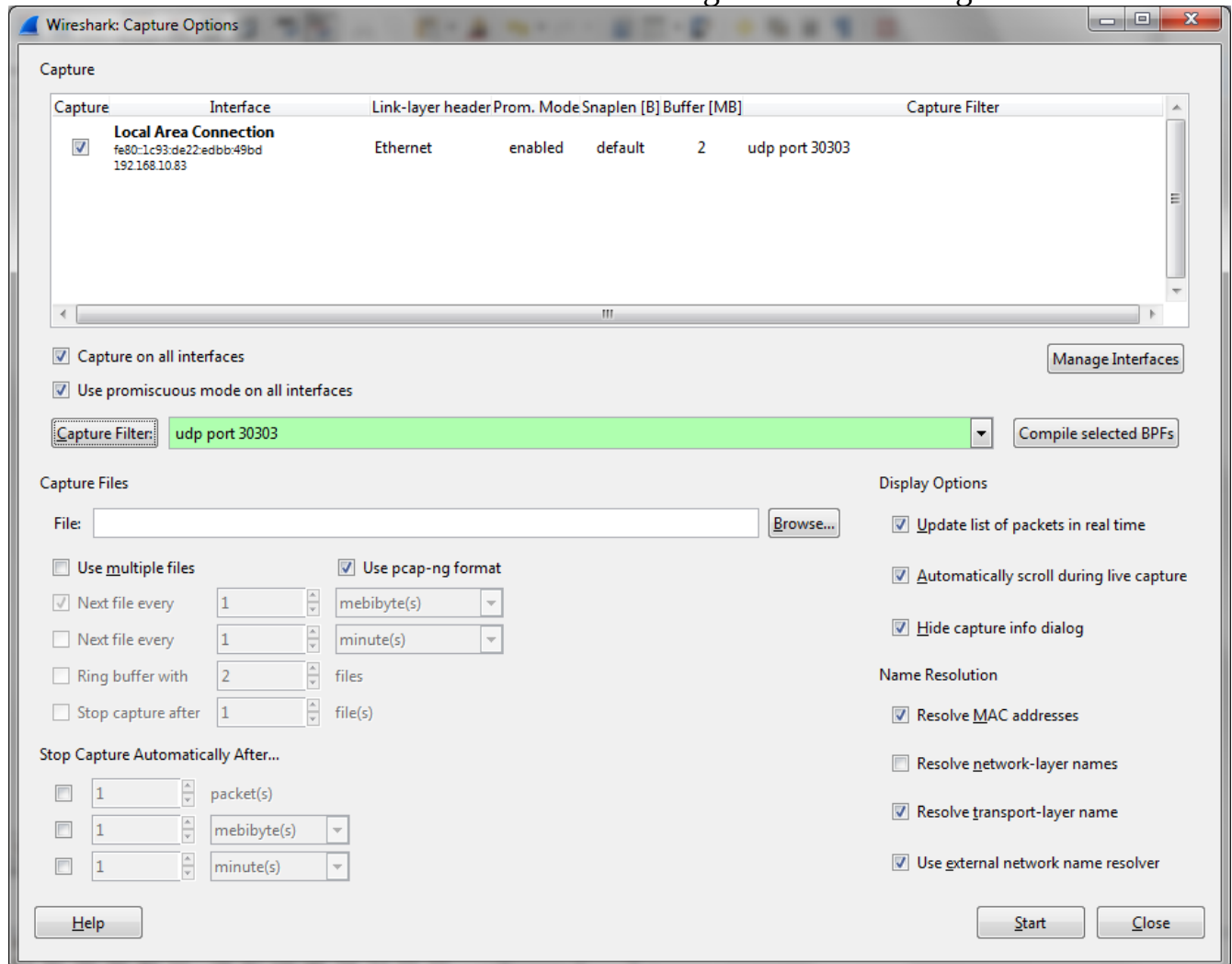
//#define TCPIP_STACK_USE_REBOOT_SERVER
#define TCPIP_STACK_USE_SNTP_CLIENT
//#define TCPIP_STACK_USE_DYNAMICDNS_CLIENT
//#define TCPIP_STACK_USE_BERKELEY_API
//#define TCPIP_STACK_USE_IPV6
#define TCPIP_STACK_USE_TCP
#define TCPIP_STACK_USE_UDP
//#define TCPIP_STACK_USE_ZEROCONF_LINK_LOCAL
//#define TCPIP_STACK_USE_ZEROCONF_MDNS_SD
#define TCPIP_STACK_COMMAND_ENABLE
//#define TCPIP_STACK_USE_IPERF
//#define TCPIP_STACK_USE_SNMP_SERVER
//#define TCPIP_STACK_USE_SNMPV3_SERVER
//#define TCPIP_STACK_USE_FTP_SERVER
//#define TCPIP_STACK_USE_DHCP_SERVER
#define TCPIP_STACK_USE_ICMP_CLIENT

```

Recompile your project and run it on the PIC32 Ethernet starter kit. Next on your PC run wireshark. Once inside wireshark click on Capture->Options menu. Click on Capture Filter and create a new like below.



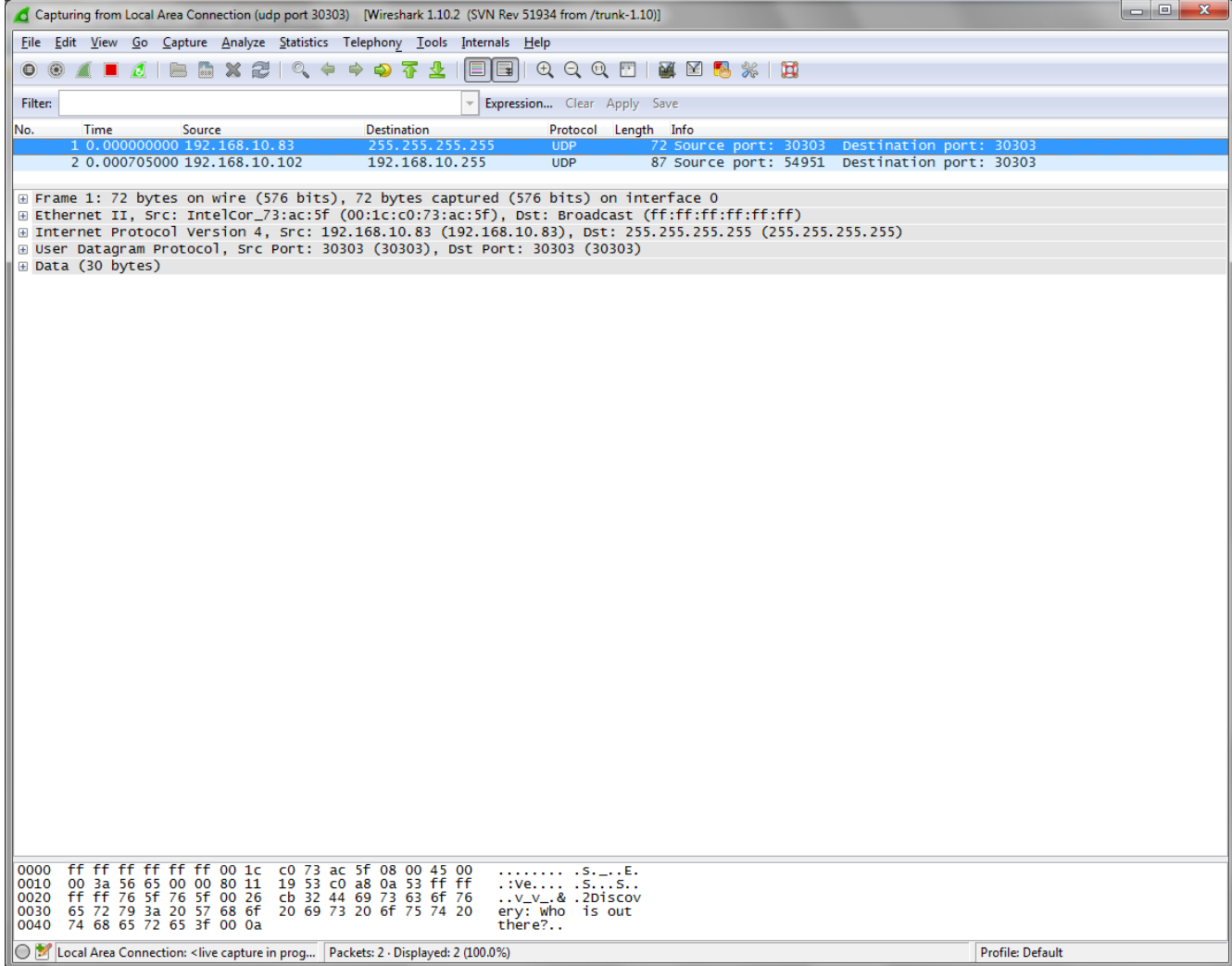
You should now have a screen that looks something like the following.



Click on the start button to start capturing packets.

Next start the tcpip_discoverer utility included in the microchip TCP/IP stack. It should be located in the directory microchip\tcpip\utilities.

Wireshark should capture two ethernet frames as shown below.



Answer the following questions about these captured frames.

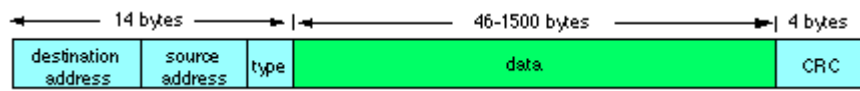
1. What is the text message contained in your UDP packets?

1. Sent from the PC?

2. Sent from the PIC?

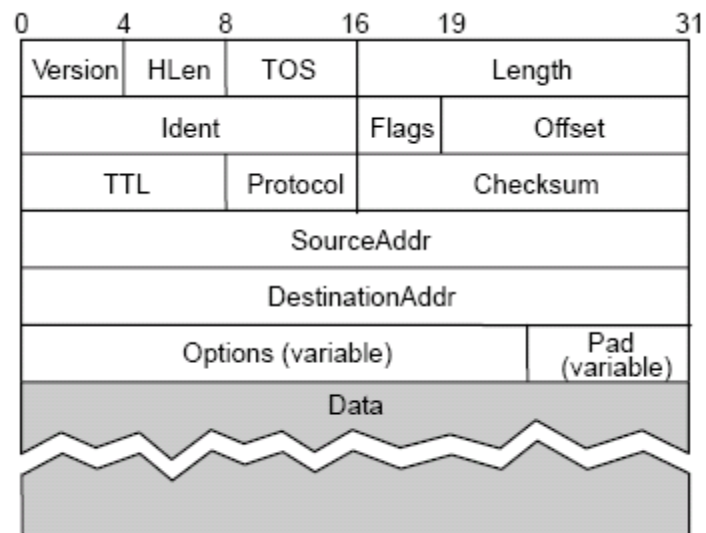
2. What is the destination IP address of the UDP packet sent from the lab computer?

3. What is the meaning of this destination IP address?



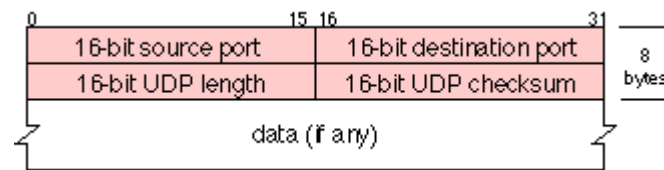
MAC encapsulation of a packet of data

4. Draw the ethernet frame for both the PC and the PIC below filling in the fields with the actual transmitted data.



14 fields comprise the IP Datagram

5. Draw the IP datagram for each the PC and the PIC filling in all fields with the actual transmitted data.



UDP Header format

6. Draw the UDP header for both the PC and the PIC filling in the values with the actual transmitted data (including actual data message sent).

Part 2

Open up a command prompt on the Lab machine and type in **telnet 10.101.0.xxx 9760**. Where xxx is the ip of your board. Once connected start typing. You should see every character you type echoed back to you in uppercase (press the escape key to end session). This is performed by the program generic_tcp_server.c. Open this file and look at the code.

Now modify this code. When the user enters a w return your name. (Demo This to the Lab Instructor)

Part 3

Open the file generic_tcp_client.c and convert all calls to *SYS_CONSOLE_MESSAGE()* to be *DBPRINTF()*. Now compile and run your code. Make sure the DBPRINTF windows is open and then press the SW2 button. Observe the results and record them below. Play with this code. We will use the TCP_Server and TCP_Client as a basis for our next lab.