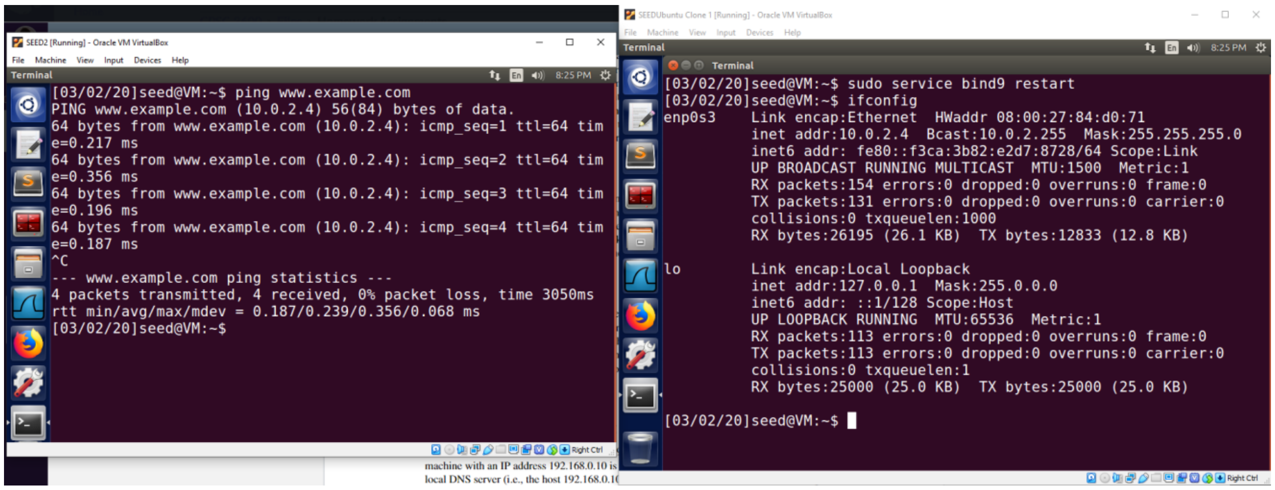
CPSC 3600 HW3

Lab Task 1

Step 1 for this project was to set up the DNS resolution so that the host machine (with the client) and the second host machine (with the server) are connected via DNS.

--


SEED2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
8:25 PM
SEEDUbuntu Clone 1 [Running] - Oracle VM VirtualBox
Right Ctrl

```
[03/02/20]seed@VM:~$ ping www.example.com
PING www.example.com (10.0.2.4) 56(84) bytes of data.
64 bytes from www.example.com (10.0.2.4): icmp_seq=1 ttl=64 tim
e=0.217 ms
64 bytes from www.example.com (10.0.2.4): icmp_seq=2 ttl=64 tim
e=0.356 ms
64 bytes from www.example.com (10.0.2.4): icmp_seq=3 ttl=64 tim
e=0.196 ms
64 bytes from www.example.com (10.0.2.4): icmp_seq=4 ttl=64 tim
e=0.187 ms
^C
--- www.example.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.187/0.239/0.356/0.068 ms
[03/02/20]seed@VM:~$
```

machine with an IP address 192.168.0.10 is
local DNS server (i.e., the host 192.168.0.1

```
[03/02/20]seed@VM:~$ sudo service bind9 restart
[03/02/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:84:d0:71
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::f3ca:3b82:e2d7:8728/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:154 errors:0 dropped:0 overruns:0 frame:0
          TX packets:131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26195 (26.1 KB)  TX bytes:12833 (12.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:113 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:25000 (25.0 KB)  TX bytes:25000 (25.0 KB)

[03/02/20]seed@VM:~$
```

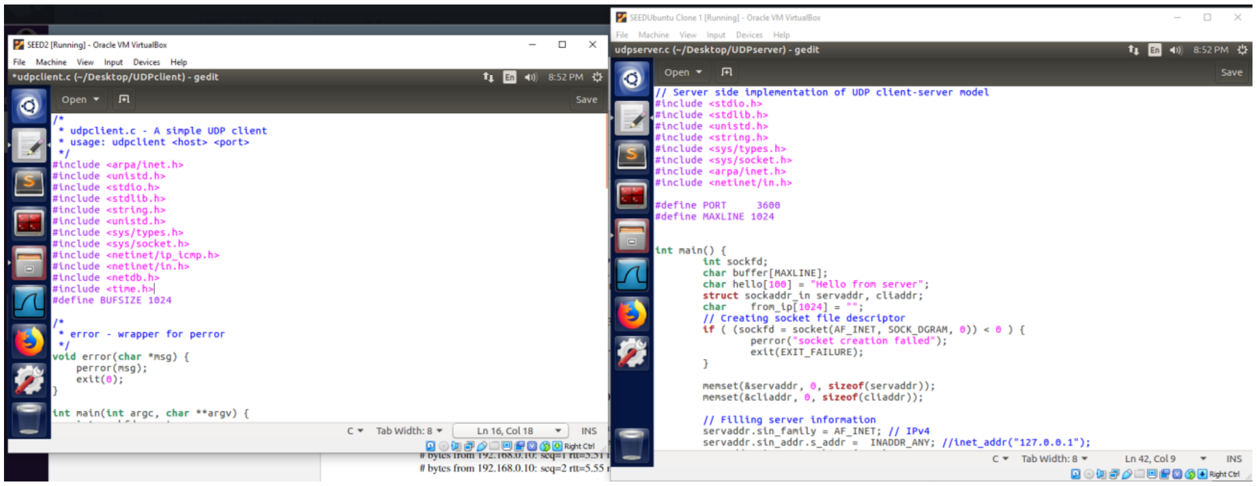
The picture above shows that www.example.com is resolved to the second host machine's IP address of 10.0.2.4.

Lab Task 2

Step 2 in this project was to setup the UDP Client on the first machine and the UDP Server on the second machine. The client used a socket type for UDP named SOCK_DGRAM and takes the hostname and port number as command line inputs. After initial setup, the client does a DNS lookup using the function gethostbyname().

For the server, the port was set to 3600.

The client was then run using www.example.com and 3600 as the command line inputs which should give access to the UDP server running on the adjacent machine.



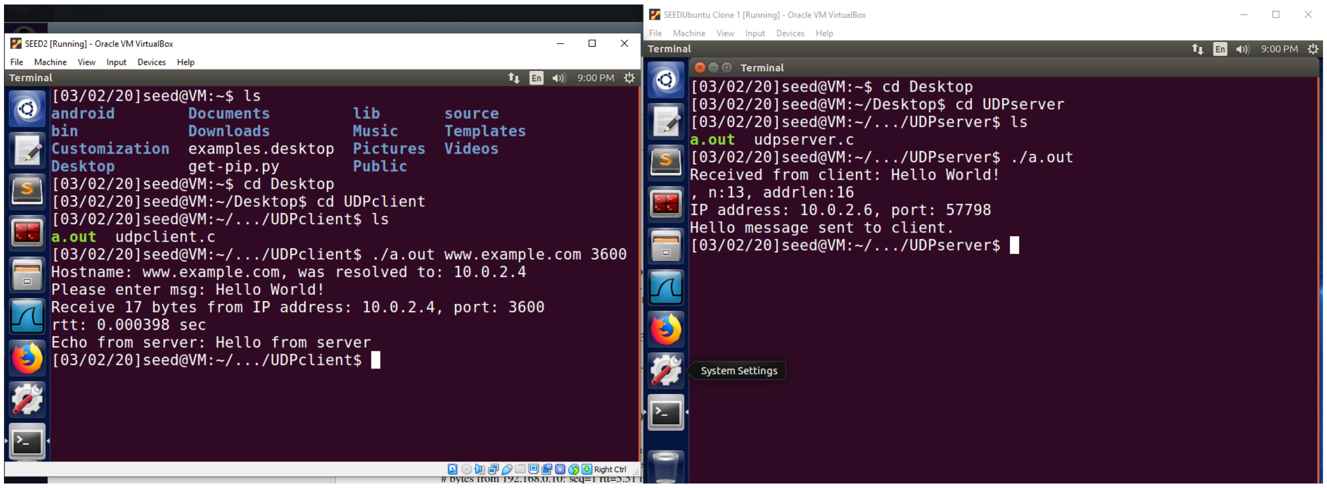
```
*udpclient.c (~/Desktop/UDPclient) - gedit
/*
 * udpclient.c - A simple UDP client
 * usage: udpclient <host> <port>
 */
#include <arpa/inet.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/ip_icmp.h>
#include <netinet/in.h>
#include <netdb.h>
#include <time.h>
#define BUFSIZE 1024

/*
 * error - wrapper for perror
 */
void error(char *msg) {
    perror(msg);
    exit(0);
}

int main(int argc, char **argv) {
```

```
udpserver.c (~/Desktop/UDPserver) - gedit
// Server side implementation of UDP client-server model
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define PORT     3600
#define MAXLINE 1024


int main() {
        int sockfd;
        char buffer[MAXLINE];
        char hello[100] = "Hello from server";
        struct sockaddr_in servaddr, cliaddr;
        char    from_ip[1024] = "";
        // Creating socket file descriptor
        if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
                perror("socket creation failed");
                exit(EXIT_FAILURE);
        }

        memset(&servaddr, 0, sizeof(servaddr));
        memset(&cliaddr, 0, sizeof(cliaddr));

        // Filling server information
        servaddr.sin_family = AF_INET; // IPv4
        servaddr.sin_addr.s_addr =  INADDR_ANY; //inet_addr("127.0.0.1");
```

The picture above shows the code for the UDP client (left) and the UDP server (right) fully setup and ready to run.

Lab Task 3

The final portion of the project was to run the server and client simultaneously, proving a connection between the two and accessing the hostname server. To do this, the UDP server was first launched on the second machine, which began waiting for the client to access it. Following this, the first machine ran the UDP client using the hostname and port as command line inputs. The client resolved the IP address correctly to the second machine's IP of 10.0.2.4. The client then sent a message to the server and the server was able to accept it and respond with the rtt and an echo of the message, as well as displaying information about the message on the second machine.



```
[03/02/20]seed@VM:~$ ls
android         Documents         lib       source
bin             Downloads         Music     Templates
Customization   examples.desktop  Pictures  Videos
Desktop         get-pip.py        Public
[03/02/20]seed@VM:~$ cd Desktop
[03/02/20]seed@VM:~/Desktop$ cd UDPclient
[03/02/20]seed@VM:~/.../UDPclient$ ls
a.out  udpclient.c
[03/02/20]seed@VM:~/.../UDPclient$ ./a.out www.example.com 3600
Hostname: www.example.com, was resolved to: 10.0.2.4
Please enter msg: Hello World!
Receive 17 bytes from IP address: 10.0.2.4, port: 3600
rtt: 0.000398 sec
Echo from server: Hello from server
[03/02/20]seed@VM:~/.../UDPclient$
```

```
[03/02/20]seed@VM:~$ cd Desktop
[03/02/20]seed@VM:~/Desktop$ cd UDPserver
[03/02/20]seed@VM:~/.../UDPserver$ ls
a.out  udpserver.c
[03/02/20]seed@VM:~/.../UDPserver$ ./a.out
Received from client: Hello World!
, n:13, addrlen:16
IP address: 10.0.2.6, port: 57798
Hello message sent to client.
[03/02/20]seed@VM:~/.../UDPserver$
```

The picture above shows the execution of the client and server communicating with one another over the resolved IP address from www.example.com to 10.0.2.4.