**Data Communication Networks – Program 2**

**Jason McQueen**

**4/17/2019**

**Documentation**

The program works as a basic copy of the ping program found in many operating systems today.

First, I made a counter to stop the loop once it pings ten times. This isn’t explicitly required in the instructions, but I thought it was better than an infinite loop. Next, it enters the infinite loop, and grabs the host name from the command line. From that, it uses a python function called getprotobyname(), which grabs the protocol’s constant from a library somewhere. Next, it tries to create a socket. If it’s unable to create a socket, it throws an error. It then prints what host name the user is attempting to connect to and tries to grab an ip address from the host name by using gethostbyname(). If it fails, it terminates the program. Next, it grabs a unique identifier from the operating system using os.getpid(). After that, it fills the ICMP packet header. I used the counter from before as the sequence number (since it is supposed to increment. After that, it uses a function called pack() to pack the icmp header and the current time stamp into a single packet. After that, it starts a clock, and times how long it takes to receive the packet. The socket has a timeout of one second, and will fail if it does not receive a reply by then. Once it grabs a reply, it prints the time it took to ping the host and sleeps for one second, and then loops back through the while loop.

Compilers:

python compiler on Ubuntu, unix environment

Development and testing:

Program was heavily tested on multiple different host names; real, fake, and local.

Sample commands that were used:

Sudo python ping\_jm4122.py google.com