CSE461 Project 3 Report

Team members:
Pingyang He
Chenghao Chuang

Description of RPC implementation:

RPCCallerSocket:
Constructor: setup tcp handler with given ip address and port number, timeout length
invoke: send handshake message using tcp handler, then send given message to the server again.

RPCService:
RegisteredHandler: Uses a map to store the servicename, method and the callback to the method.
Server(ServerSocket) extends Thread: This class creates a thread and takes over the server socket to wait for the coming connections. Whenever a connection is accepted, it forks a new thread to handle the handshake and request.
RPCHandler extends Thread: This class handles the handshake and the invoke requests. It initiates a TCPMessageHandler and uses it to send/receive messages. It follows the rpc protocol, but if the handshake/invoke message is not following the rpc protocol, it would return an error response to the caller.

What works:

we can ping and echo to another server, and get correct respond back
Our RPCService can receive multiple Pings from other clients.
Our Android app can ping other servers, and respond to other's ping/echo correctly.

What doesn't work:

N/A

Ping to server:

```
 OS  edu.uw.cs.cse461.sp12.OS.RPCService started
 OS  edu.uw.cs.cse461.sp12.OS.EchoService started
Enter app name (whoami  echo  ping  ), or exit: ping
Enter lines like <target> <msg> to have <msg> echoed back
Enter a host ip, or exit to exit: cse461.cs.washington.edu
Enter the RPC port, or empty line to exit: 46120
Test 1: IP=cse461.cs.washington.edu host=46120 time=0.511654664 seconds
Test 2: IP=cse461.cs.washington.edu host=46120 time=0.463847769 seconds
Test 3: IP=cse461.cs.washington.edu host=46120 time=0.445056636 seconds
Test 4: IP=cse461.cs.washington.edu host=46120 time=0.453924096 seconds
Test 5: IP=cse461.cs.washington.edu host=46120 time=0.455499663 seconds
Enter a host ip, or exit to exit:
```

Native Ping to server:

```
Pinging cse461.cs.washington.edu [128.208.2.207] with 32 bytes of data:
Reply from 128.208.2.207: bytes=32 time=77ms TTL=63
Reply from 128.208.2.207: bytes=32 time=1ms TTL=63
```

Reply from 128.208.2.207: bytes=32 time=2ms TTL=63
Reply from 128.208.2.207: bytes=32 time=3ms TTL=63

Ping statistics for 128.208.2.207:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 77ms, Average = 20ms