Communications Node

December, 15th 2016

Andrei Polzounov

Overview

The communication node program allows seeing other communication nodes running on the same local area network. It uses UDP multicast messages (https://tools.ietf.org/html/rfc1112) for one-to-many communication and TCP messages for one-to-one communication. The implementation is done in C++11 with boost::asio as the main networking library. The implementation is primarily Linux based, but has also been tested on Windows. It allows for an arbitrary number of CNs and multiple CNs can run on the same machine.

Program Design

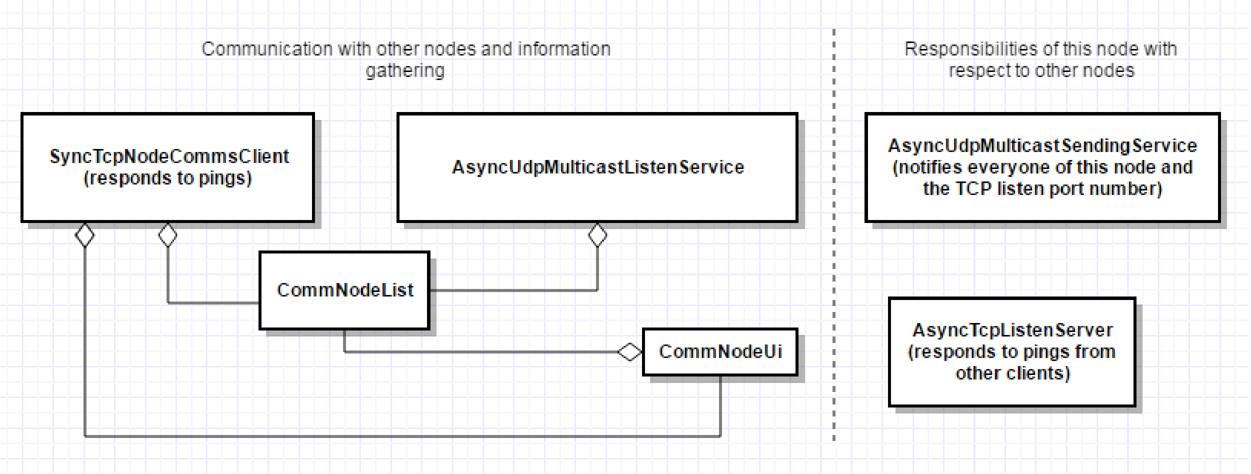

Communication with other nodes and information gathering
Responsibilities of this node with respect to other nodes
SyncTcpNodeCommsClient (responds to pings)
AsyncUdpMulticastListenService
AsyncUdpMulticastSendingService (notifies everyone of this node and the TCP listen port number)
CommNodeList
CommNodeUi
AsyncTcpListenServer (responds to pings from other clients)

Figure 1: Class diagram

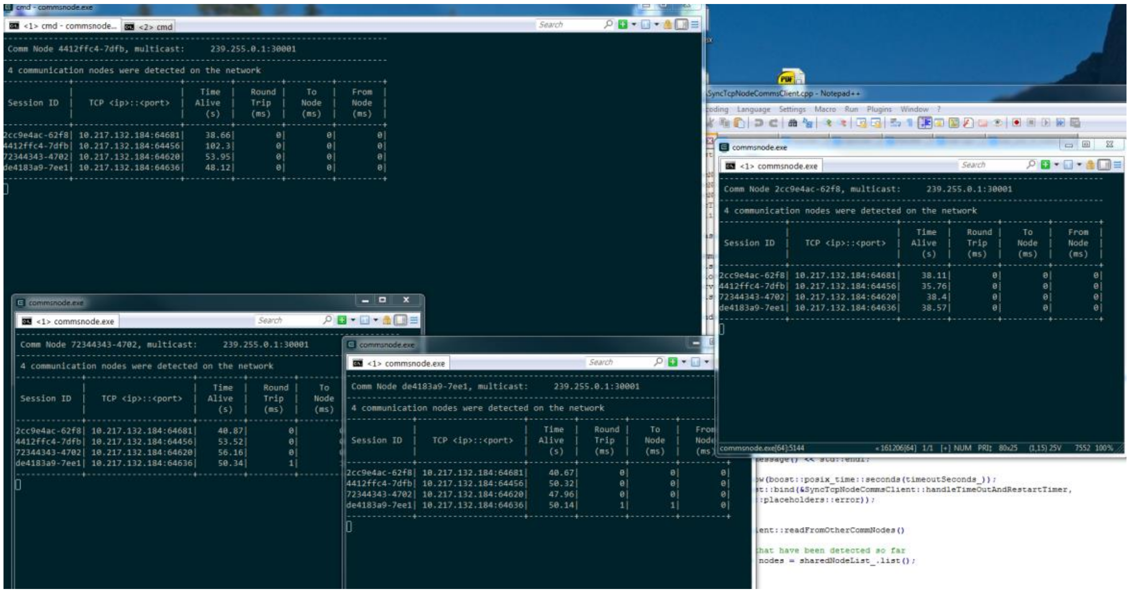
The main design consists of six classes:

1. AsyncUdpMulticastSendingService – class that is responsible for sending the information about this communication node to others. The information is encoded as a JSON string in the following format:

```
{
  "SessionId": "e89a00ca-fa42-432c-8ce3-8149ea935b25",
  "TcpServerPort": "54829"
}
```

   The session ID is a unique identifier attributed to each CN, the TCP server port is a port assigned by the OS to this CN.

2. AsyncUdpMulticastListenService – other nodes are responsible for listening to incoming multicast, parsing the JSON and storing the session IDs and ports into a shared list.
3. CommNodeList – this is a shared directory for different threads to be able to add new CNs, delete non-responding CNs and finally to present them to the user.
4. AsyncTcpListenServer – this is a server with the TcpServerPort number referenced above. This asynchronous server responds to queries from other nodes.
5. SyncTcpNodeCommsClient – this is the only synchronous communications class. It uses blocking calls to try to measure the round-trip-time for each query and response.
6. CommNodeUi – this class presents the information to the user. In Linux this relies on ncurses and in Windows uses a system(“clr”) to present a continuous table.



```
--------------------------------------------------------------------------
 Comm Node 4412ffc4-7dfb, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 4 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |   Time  |  Round  |    To   |   From  |
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node  |   Node  |
             |                     |    (s)  |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
2cc9e4ac-62f8| 10.217.132.184:64681|    38.66|        0|        0|        0|
4412ffc4-7dfb| 10.217.132.184:64456|    102.3|        0|        0|        0|
72344343-4702| 10.217.132.184:64620|    53.95|        0|        0|        0|
de4183a9-7ee1| 10.217.132.184:64636|    48.12|        0|        0|        0|
-------------+---------------------+---------+---------+---------+---------+
```

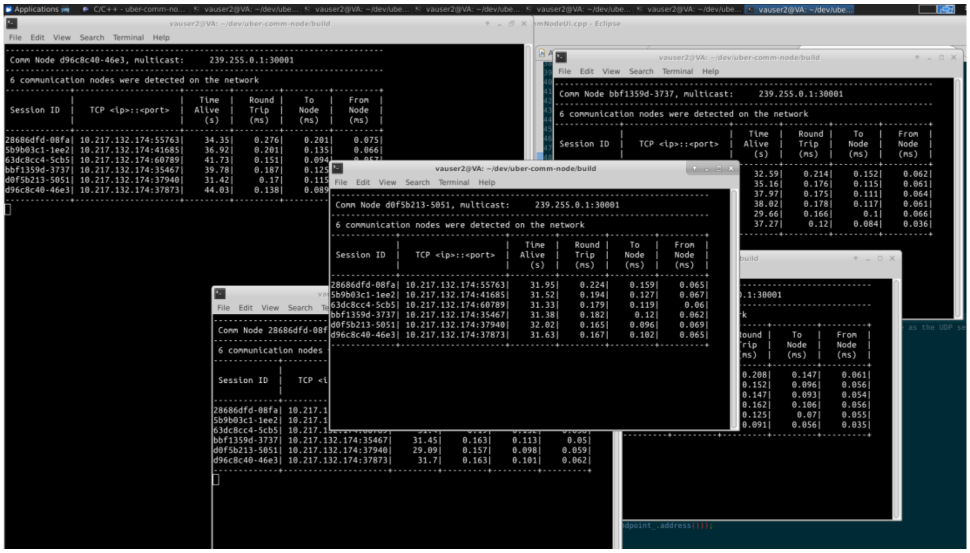
```
--------------------------------------------------------------------------
 Comm Node 2cc9e4ac-62f8, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 4 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |   Time  |  Round  |    To   |   From  |
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node  |   Node  |
             |                     |    (s)  |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
2cc9e4ac-62f8| 10.217.132.184:64681|    38.11|        0|        0|        0|
4412ffc4-7dfb| 10.217.132.184:64456|    35.76|        0|        0|        0|
72344343-4702| 10.217.132.184:64620|     38.4|        0|        0|        0|
de4183a9-7ee1| 10.217.132.184:64636|    38.57|        0|        0|        0|
-------------+---------------------+---------+---------+---------+---------+
```

```
--------------------------------------------------------------------------
 Comm Node 72344343-4702, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 4 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------
             |                     |   Time  |  Round  |    To
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node
             |                     |    (s)  |  (ms)   |   (ms)
-------------+---------------------+---------+---------+---------
2cc9e4ac-62f8| 10.217.132.184:64681|    40.87|        0|
4412ffc4-7dfb| 10.217.132.184:64456|    53.52|        0|
72344343-4702| 10.217.132.184:64620|    56.16|        0|
de4183a9-7ee1| 10.217.132.184:64636|    50.34|        1|
-------------+---------------------+---------+---------+---------
```

```
--------------------------------------------------------------------------
 Comm Node de4183a9-7ee1, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 4 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |   Time  |  Round  |    To   |   From  |
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node  |   Node  |
             |                     |    (s)  |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
2cc9e4ac-62f8| 10.217.132.184:64681|    40.67|        0|        0|        0|
4412ffc4-7dfb| 10.217.132.184:64456|    50.32|        0|        0|        0|
72344343-4702| 10.217.132.184:64620|    47.96|        0|        0|        0|
de4183a9-7ee1| 10.217.132.184:64636|    50.14|        1|        1|        0|
-------------+---------------------+---------+---------+---------+---------+
```

cmd - commsnode.exe
<1> cmd - commsnode...
<2> cmd
commsnode.exe
<1> commsnode.exe
Search
SyncTcpNodeCommsClient.cpp - Notepad++
Coding Language Settings Macro Run Plugins Window ?
commsnode.exe[64]:5144 «161206[64] 1/1 [+] NUM PRI‡ 80x25 (1,15) 25V 7552 100%

Figure 2: Windows implementation running several comm nodes



```
--------------------------------------------------------------------------
 Comm Node d96c8c40-46e3, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 6 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |   Time  |  Round  |    To   |   From  |
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node  |   Node  |
             |                     |    (s)  |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
28686dfd-08fa| 10.217.132.174:55763|    34.35|    0.276|    0.201|    0.075|
5b9b03c1-1ee2| 10.217.132.174:41685|    36.92|    0.201|    0.135|    0.066|
63dc8cc4-5cb5| 10.217.132.174:60789|    41.73|    0.151|    0.094|
bbf1359d-3737| 10.217.132.174:35467|    39.78|    0.187|    0.125
d0f5b213-5051| 10.217.132.174:37940|    31.42|     0.17|    0.115
d96c8c40-46e3| 10.217.132.174:37873|    44.03|    0.138|    0.089
-------------+---------------------+---------+---------+-----
```

```
--------------------------------------------------------------------------
 Comm Node bbf1359d-3737, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 6 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |   Time  |  Round  |    To   |   From  |
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node  |   Node  |
             |                     |    (s)  |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
                                   |    32.59|    0.214|    0.152|    0.062|
                                   |    35.16|    0.176|    0.115|    0.061|
                                   |    37.97|    0.175|    0.111|    0.064|
                                   |    38.02|    0.178|    0.117|    0.061|
                                   |    29.66|    0.166|      0.1|    0.066|
                                   |    37.27|     0.12|    0.084|    0.036|
-------------+---------------------+---------+---------+---------+---------+
```

```
--------------------------------------------------------------------------
 Comm Node d0f5b213-5051, multicast:     239.255.0.1:30001
--------------------------------------------------------------------------
 6 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |   Time  |  Round  |    To   |   From  |
 Session ID  |   TCP <ip>::<port>  |  Alive  |  Trip   |   Node  |   Node  |
             |                     |    (s)  |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
28686dfd-08fa| 10.217.132.174:55763|    31.95|    0.224|    0.159|    0.065|
5b9b03c1-1ee2| 10.217.132.174:41685|    31.52|    0.194|    0.127|    0.067|
63dc8cc4-5cb5| 10.217.132.174:60789|    31.33|    0.179|    0.119|     0.06|
bbf1359d-3737| 10.217.132.174:35467|    31.38|    0.182|     0.12|    0.062|
d0f5b213-5051| 10.217.132.174:37940|    32.02|    0.165|    0.096|    0.069|
d96c8c40-46e3| 10.217.132.174:37873|    31.63|    0.167|    0.102|    0.065|
-------------+---------------------+---------+---------+---------+---------+
```

```
--------------------------------------------------------------------------
 Comm Node 28686dfd-08f
--------------------------------------------------------------------------
 6 communication nodes
-------------+---------------------+---------+---------+---------+---------+
             |
 Session ID  |   TCP <i
             |
-------------+---------------------+---------+---------+---------+---------+
28686dfd-08fa| 10.217.1
5b9b03c1-1ee2| 10.217.1
63dc8cc4-5cb5| 10.217.1
bbf1359d-3737| 10.217.132.174:35467|    31.45|    0.163|    0.113|     0.05|
d0f5b213-5051| 10.217.132.174:37940|    29.09|    0.157|    0.098|    0.059|
d96c8c40-46e3| 10.217.132.174:37873|     31.7|    0.163|    0.101|    0.062|
-------------+---------------------+---------+---------+---------+---------+
```

```
                 0.1:30001
--------------------------------------------------------------------------
                 k
-------------+---------------------+---------+---------+---------+---------+
                                             |  Round  |    To   |   From  |
                                             |  Trip   |   Node  |   Node  |
                                             |  (ms)   |   (ms)  |   (ms)  |
-------------+---------------------+---------+---------+---------+---------+
                                             |    0.208|    0.147|    0.061|
                                             |    0.152|    0.096|    0.056|
                                             |    0.147|    0.093|    0.054|
                                             |    0.162|    0.106|    0.056|
                                             |    0.125|     0.07|    0.055|
                                             |    0.091|    0.056|    0.035|
-------------+---------------------+---------+---------+---------+---------+
```

Figure 3: Linux implementation running several comm nodes

Build instructions

Linux

In a Linux environment – make sure that git, boost, cmake and ncurses are installed. In Ubuntu:

```
sudo apt-get install libboost-all-dev libncurses5-dev
```

From the main directory - move to the build directory and then run cmake in it. The CMakeLists.txt script will automatically git-clone and build googletest as part of the build process.

```
cd build
cmake ..
make
# run the program
./commsnode --help
```

Windows

Dependencies:

1) MinGW compiler suite with libboost: https://nuwen.net/mingw.html unzip it to C:\MinGW and ensure that "C:\MinGW\bin;C:\MinGW\git\bin" is part of Windows %PATH%

The build can be done from the main directory with a simple Makefile – an example is provided in Makefile.windows

```
copy Makefile.windows Makefile
make
cd build
# run the program
Commsnode.exe --help
```

Unit Tests

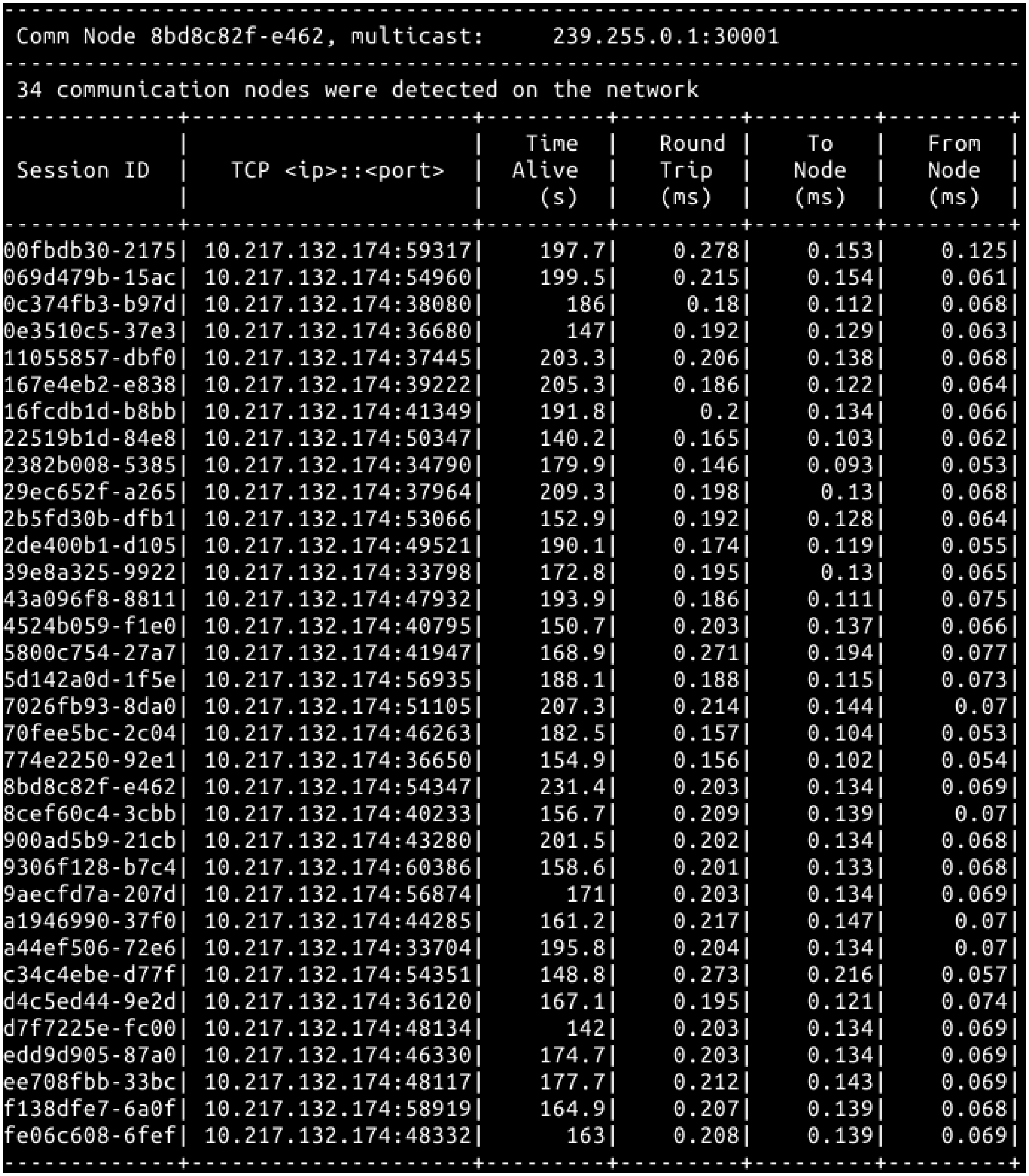
The unit tests will be built automatically on Linux. The cmake script will clone the latest googletest master from GitHub and build it in place. Following that the unit tests will be built and can be run with

```
./test_commsnode
```

The googletest cmake script is based on the following GitHub repository - https://github.com/snikulov/google-test-examples.

Networking Tools

In the tools directory two debugging tools are provided rx_udp and rx_tcp. They can be built using the Makefile.linux and Makefile.windows examples in the directory. These tools can be used to analyze the TCP and UDP messages being sent across the network.



```
----------------------------------------------------------------------------
 Comm Node 8bd8c82f-e462, multicast:     239.255.0.1:30001
----------------------------------------------------------------------------
 34 communication nodes were detected on the network
-------------+---------------------+---------+---------+---------+---------+
             |                     |  Time   |  Round  |   To    |  From   |
 Session ID  |  TCP <ip>::<port>   |  Alive  |  Trip   |  Node   |  Node   |
             |                     |   (s)   |  (ms)   |  (ms)   |  (ms)   |
-------------+---------------------+---------+---------+---------+---------+
00fbdb30-2175| 10.217.132.174:59317|    197.7|    0.278|    0.153|    0.125|
069d479b-15ac| 10.217.132.174:54960|    199.5|    0.215|    0.154|    0.061|
0c374fb3-b97d| 10.217.132.174:38080|      186|     0.18|    0.112|    0.068|
0e3510c5-37e3| 10.217.132.174:36680|      147|    0.192|    0.129|    0.063|
11055857-dbf0| 10.217.132.174:37445|    203.3|    0.206|    0.138|    0.068|
167e4eb2-e838| 10.217.132.174:39222|    205.3|    0.186|    0.122|    0.064|
16fcdb1d-b8bb| 10.217.132.174:41349|    191.8|      0.2|    0.134|    0.066|
22519b1d-84e8| 10.217.132.174:50347|    140.2|    0.165|    0.103|    0.062|
2382b008-5385| 10.217.132.174:34790|    179.9|    0.146|    0.093|    0.053|
29ec652f-a265| 10.217.132.174:37964|    209.3|    0.198|     0.13|    0.068|
2b5fd30b-dfb1| 10.217.132.174:53066|    152.9|    0.192|    0.128|    0.064|
2de400b1-d105| 10.217.132.174:49521|    190.1|    0.174|    0.119|    0.055|
39e8a325-9922| 10.217.132.174:33798|    172.8|    0.195|     0.13|    0.065|
43a096f8-8811| 10.217.132.174:47932|    193.9|    0.186|    0.111|    0.075|
4524b059-f1e0| 10.217.132.174:40795|    150.7|    0.203|    0.137|    0.066|
5800c754-27a7| 10.217.132.174:41947|    168.9|    0.271|    0.194|    0.077|
5d142a0d-1f5e| 10.217.132.174:56935|    188.1|    0.188|    0.115|    0.073|
7026fb93-8da0| 10.217.132.174:51105|    207.3|    0.214|    0.144|     0.07|
70fee5bc-2c04| 10.217.132.174:46263|    182.5|    0.157|    0.104|    0.053|
774e2250-92e1| 10.217.132.174:36650|    154.9|    0.156|    0.102|    0.054|
8bd8c82f-e462| 10.217.132.174:54347|    231.4|    0.203|    0.134|    0.069|
8cef60c4-3cbb| 10.217.132.174:40233|    156.7|    0.209|    0.139|     0.07|
900ad5b9-21cb| 10.217.132.174:43280|    201.5|    0.202|    0.134|    0.068|
9306f128-b7c4| 10.217.132.174:60386|    158.6|    0.201|    0.133|    0.068|
9aecfd7a-207d| 10.217.132.174:56874|      171|    0.203|    0.134|    0.069|
a1946990-37f0| 10.217.132.174:44285|    161.2|    0.217|    0.147|     0.07|
a44ef506-72e6| 10.217.132.174:33704|    195.8|    0.204|    0.134|     0.07|
c34c4ebe-d77f| 10.217.132.174:54351|    148.8|    0.273|    0.216|    0.057|
d4c5ed44-9e2d| 10.217.132.174:36120|    167.1|    0.195|    0.121|    0.074|
d7f7225e-fc00| 10.217.132.174:48134|      142|    0.203|    0.134|    0.069|
edd9d905-87a0| 10.217.132.174:46330|    174.7|    0.203|    0.134|    0.069|
ee708fbb-33bc| 10.217.132.174:48117|    177.7|    0.212|    0.143|    0.069|
f138dfe7-6a0f| 10.217.132.174:58919|    164.9|    0.207|    0.139|    0.068|
fe06c608-6fef| 10.217.132.174:48332|      163|    0.208|    0.139|    0.069|
-------------+---------------------+---------+---------+---------+---------+
```

Figure 4: Many nodes running on Linux

Discussion

Boost::asio is a very nice library for network programming. It made it very easy to set up asynchronous listening services. Asynchronisity is important for avoiding deadlocks. In fact the only synchronous that is used in this implementation is the SyncTcpNodeCommsClient, because it needs to measure round-trip-time between querying and receiving a response from the other nodes. This is the main weak point of the program – because if the Client hangs the UI will not be updated until TCP times out.

The other main drawback of using this library for a small project is that I did not have enough time to properly wrap or encapsulate the network I/O calls. Doing so would have allowed me to create mock classes (googlemock) and I could have written unit tests for all of the network facing services. As it is only the classes that are completely internal to this application have unit tests.

Another drawback, is that while the round-trip-time is somewhat accurate (minus some time spent in the application layer), the estimated to and from times are relying on another CNs clock. This is problematic because the other clock might not be synced. Ideally this requires something like the IEEE 1588-2002 (Precision Time Protocol) to synchronize all of the clocks on the network.

Overall, it is a good solution given the time given – it does not require any manual settings changes (even though settings can be provided from the command line). It uses UDP multicast to broadcast the TCP ports to allow for a TCP connection and it can recover from non-responsive nodes. If I had more time I would have come up with a way to asynchronize the round-trip timing information and with a metric for measuring node connectivity.