Programming in RMI & UDP:
Ease of use and relative reliability

Nathan Amar, *EIE Undergraduate*, *Imperial College*
Raphael Bijaoui, *EIE Undergraduate*, *Imperial College*

The purpose of this report was to explore the development of two client-server interactions, using the User Data Protocol (UDP) and then the Java Remote Method Invocation (RMI) API. Following the implementation of these two different technologies, we were able to assess their relative reliability and ease of use for effective future application.

Reliability

Message Loss

The RMI mechanism successfully transmitted each message in all the test cases.
For example, as we increased our messages from 200 - 2000, we observed no message loss on our Client-Server interaction (Appendix B). This is because RMI follows the TCP/IP protocol, which detects packet loss and performs retransmissions for reliable messaging.

Contrarily, the UDP mechanism was not as reliable in transmitting the messages. While transmitting 200 to 1400 messages of size 3000 bytes, none were lost. But starting at 1400 sent, we began to observe data loss, which increased exponentially (Appendix B).
There are several factors that may contribute to this problem. We primarily observed how different packet sizes affects the reliability of the transmission. (Appendix C) clearly showed us that as the packet size increased, our number of lost messages increased linearly. Running sysctl on our server machine, we observed that the maximum space for incoming UDP datagrams had a size of 787kB (Appendix F). This helps explain why as we approached this limit by increasing our packet size, we observed more system malfunctioning, as our buffer remained filled with data while incoming messages arrived for processing. Other possible factors for UDP packet loss may include but are not limited to- network hardware/driver failure, and signal attenuation due to interference from our environment.

Time

As expected, both RMI and UDP take longer to send/receive all messages as the number of messages increases. However, the time required for RMI was observed to be of much greater order than UDP (Appendix E). This is because as opposed to UDP, the RMI first sets up a connection establishment and then sends all messages directly to the server. It then retransmits any lost packets and returns the respective receipt of the messages in sequential order. This requires more time than simply transmitting messages to the server regardless of outcome, as done in UDP. We also observed that both were sensitive to the physical distance between client & server. It was clear that having the machines beside each other generally provided the best results (Appendix E).

Ease of use

While conceptually the RMI mechanism was easier to program, we actually found that implementing it was more difficult than UDP. This was due to the fact that RMI works at a higher level of abstraction than UDP, with premade libraries for Naming, Remote Exceptions, server object creation, and most usefully- registry management. This reduced the number of lines in our program, however we found it harder to debug. We particularly had problems with the Security Manager. When compiling, an error persisted, and we weren't sure where in the data transmission process our problem persisted. Since we were further removed from our actual code, it was not clear what exactly was the issue with the security permissions in our application. It was only after investigation that we discovered the RMISecurityManager was deprecated, encouraging us to fully implement it using System.SecurityManager

Appendix A: program execution in terminal

1. UDP Server Packet Loss

```
(base) nathanamar@dyn3171-82:$ ./udpserver.sh 1099  986896
UDPServer ready
Receiving messages
Messages have been processed...
250/1000         Successfully Received
750/1000                 Messages Lost
Lost packet numbers:  215,  216,  217,  218,  219,  220,  222,  223,  224,  225,  226,  227,  228,  229,  230,  231,  2
32,  233,  234,  235,  236,  237,  238,  239,  240,  241,  242,  243,  244,  245,  246,  247,  248,  249,  250,  251,
252,  253,  254,  255,  256,  257,  258,  259,  260,  261,  262,  263,  264,  265,  266,  267,  268,  269,  270,  271,
 272,  273,  274,  275,  276,  277,  278,  279,  280,  281,  282,  283,  284,  286,  287,  288,  289,  290,  291,  292,
  293,  294,  295,  296,  297,  298,  299,  300,  301,  302,  303,  304,  305,  306,  307,  308,  309,  310,  311,  312
,  313,  314,  315,  316,  317,  318,  319,  320,  321,  322,  323,  324,  325,  326,  327,  328,  329,  330,  331,  33
2,  333,  334,  335,  336,  337,  338,  339,  340,  341,  342,  343,  344,  345,  346,  347,  348,  351,  352,  353,  3
54,  355,  356,  357,  358,  359,  360,  361,  362,  363,  364,  365,  366,  367,  368,  369,  370,  371,  372,  373,
374,  375,  376,  377,  378,  379,  380,  381,  382,  383,  384,  385,  386,  387,  388,  389,  390,  391,  392,  393,
 394,  395,  396,  397,  398,  399,  400,  401,  402,  403,  404,  405,  406,  407,  408,  409,  410,  411,  416,  417,
  418,  419,  420,  421,  422,  423,  424,  425,  426,  427,  428,  429,  430,  431,  432,  433,  434,  435,  436,  437
,  438,  439,  440,  441,  442,  443,  444,  445,  446,  447,  448,  449,  450,  451,  452,  453,  454,  455,  456,  45
7,  458,  459,  460,  461,  462,  463,  464,  465,  466,  467,  468,  469,  470,  471,  472,  473,  474,  475,  476,  4
77,  478,  479,  480,  481,  482,  483,  484,  485,  486,  487,  488,  489,  490,  491,  492,  493,  494,  495,  496,
497,  498,  499,  500,  501,  502,  503,  504,  505,  506,  507,  508,  509,  510,  511,  512,  513,  514,  515,  516,
 517,  518,  519,  520,  521,  522,  523,  524,  525,  526,  527,  528,  529,  530,  531,  532,  533,  534,  535,  536,
  537,  538,  539,  540,  541,  542,  543,  544,  545,  546,  547,  548,  549,  550,  551,  552,  553,  554,  555,  556
,  557,  558,  559,  561,  562,  563,  564,  565,  566,  567,  568,  569,  570,  571,  572,  573,  574,  575,  576,  57
7,  578,  579,  580,  581,  582,  583,  584,  585,  586,  587,  588,  589,  590,  591,  592,  593,  594,  595,  596,  5
97,  598,  599,  600,  601,  602,  603,  604,  605,  606,  607,  608,  609,  610,  611,  612,  613,  614,  615,  616,
617,  618,  619,  620,  621,  623,  624,  625,  626,  627,  628,  629,  630,  631,  632,  633,  634,  635,  636,  637,
 638,  639,  640,  641,  642,  643,  644,  645,  646,  647,  648,  649,  650,  651,  652,  653,  654,  655,  656,  657,
  658,  659,  660,  661,  662,  663,  664,  665,  666,  667,  668,  669,  670,  671,  672,  673,  674,  675,  676,  677
,  678,  679,  680,  681,  682,  683,  684,  685,  686,  687,  688,  689,  690,  691,  692,  693,  694,  695,  696,  69
7,  698,  699,  700,  701,  702,  703,  704,  705,  706,  707,  708,  709,  710,  711,  712,  713,  714,  715,  716,  7
17,  718,  719,  720,  721,  722,  723,  724,  725,  726,  727,  728,  729,  730,  731,  732,  733,  734,  735,  736,
737,  738,  739,  740,  741,  742,  743,  744,  745,  746,  747,  748,  749,  750,  751,  752,  753,  754,  755,  756,
 757,  758,  759,  760,  761,  762,  763,  764,  765,  766,  767,  768,  769,  770,  771,  772,  773,  774,  775,  776,
```

```
(base) nathanamar@dyn3171-82:$ ./udpserver.sh 1099 3000
UDPServer ready
Receiving messages
Messages have been processed...
100/100          Successfully Received
0/100            Messages Lost
Lost packet numbers: None
Done
(base) nathanamar@dyn3171-82:$ ./udpserver.sh 1099 3000
UDPServer ready
Receiving messages
Messages have been processed...
300/300          Successfully Received
0/300            Messages Lost
Lost packet numbers: None
Done
(base) nathanamar@dyn3171-82:$ ./udpserver.sh 1099 3000
UDPServer ready
Receiving messages
Messages have been processed...
500/500          Successfully Received
0/500            Messages Lost
Lost packet numbers: None
Done
```

2. UDP Client

```
na6518@skee-304-015:~/Documents/given templates$ ./udpclient.sh 146.169.175.37 1099 100
Elapsed Time: 20
na6518@skee-304-015:~/Documents/given templates$ ./udpclient.sh 146.169.175.37 1099 300
Elapsed Time: 22
na6518@skee-304-015:~/Documents/given templates$ ./udpclient.sh 146.169.175.37 1099 500
Elapsed Time: 23
```

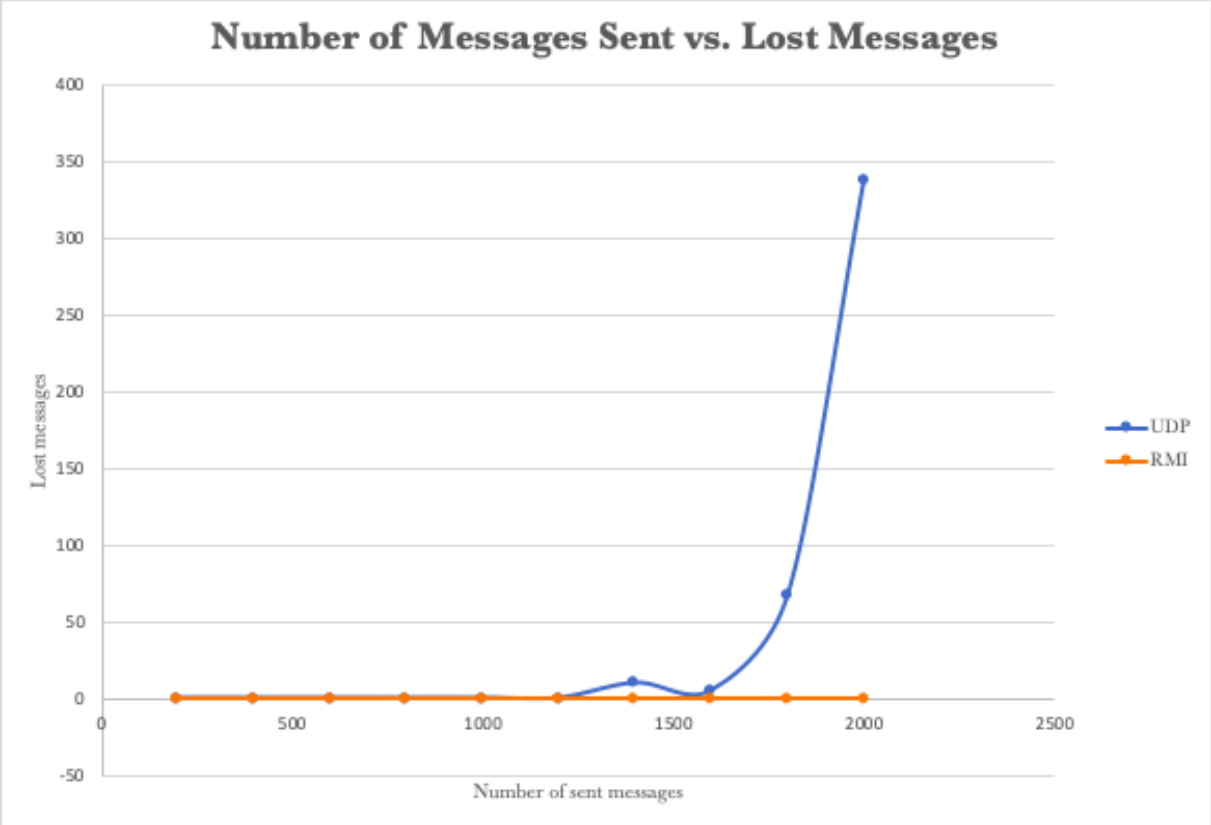
3. RMI Server

```
dyn3139-160:CW raphaelbijaoui$ ./rmiserver.sh
Running
ALL 100/100 messages received! :)
```

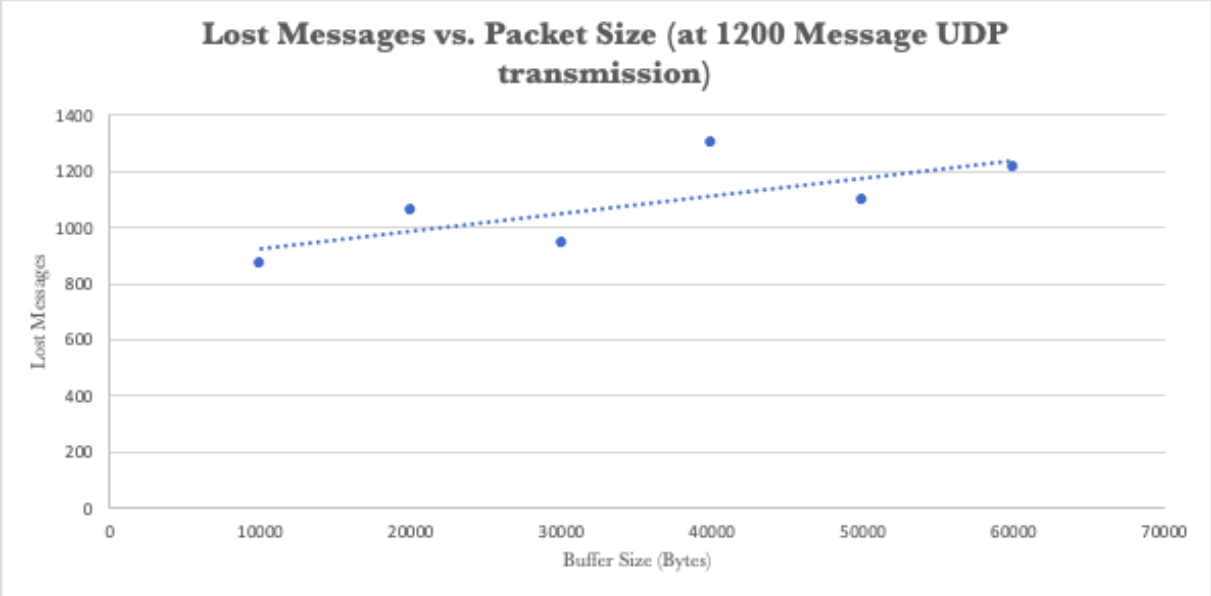
4. RMI Client

```
na6518@skee-304-015:~/Documents/given templates$ ./rmiclient.sh 146.169.175.37 300
Elapsed Time: 1736
na6518@skee-304-015:~/Documents/given templates$ ./rmiclient.sh 146.169.175.37 200
Elapsed Time: 1045
```

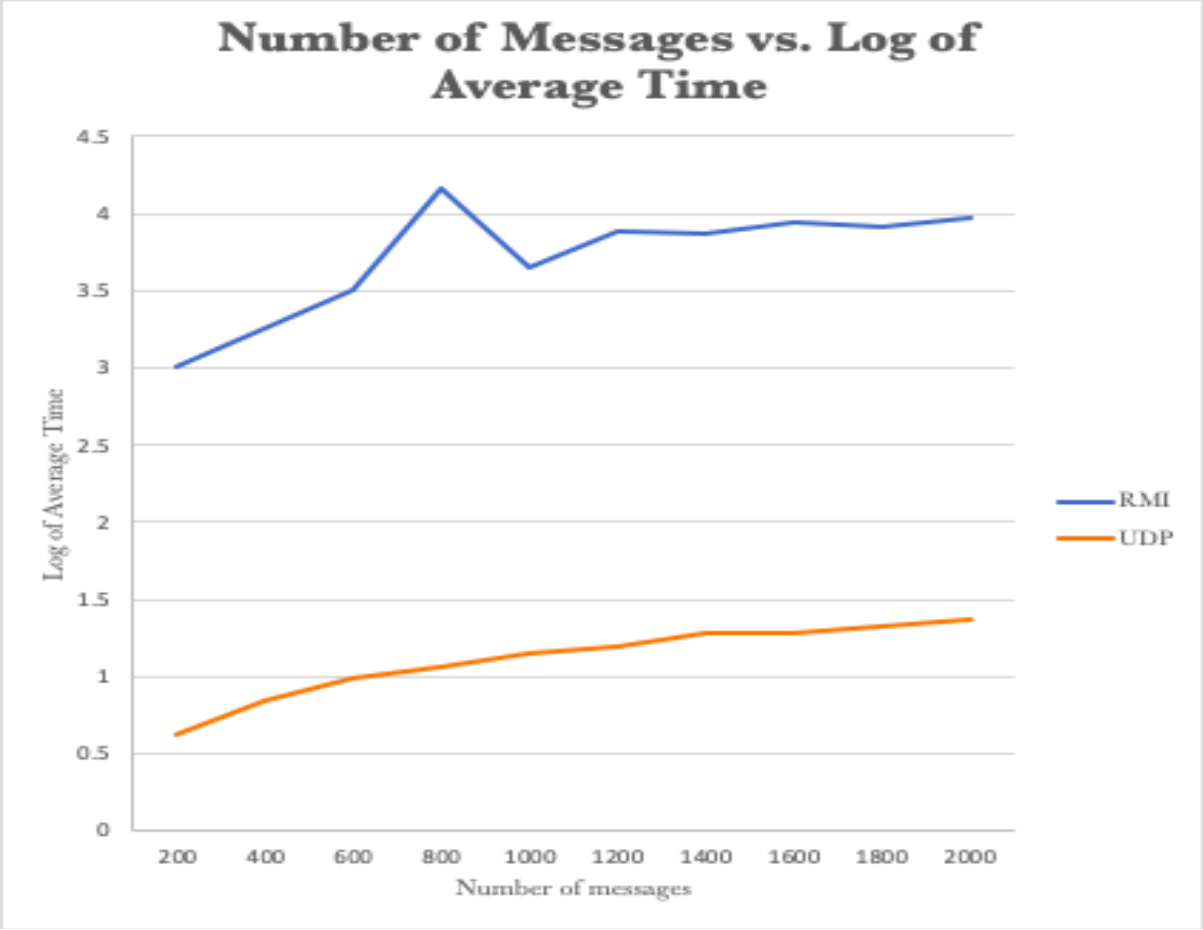
Appendix B



Appendix C



Appendix D



Appendix E

Appendix F: Max UDP buffer space

```
dyn3139-160:~ raphaelbijaoui$ sysctl -a
user.cs_path: /usr/bin:/bin:/usr/sbin:/sbin
user.bc_base_max: 99
user.bc_dim_max: 2048
user.bc_scale_max: 99
user.bc_string_max: 1000
user.coll_weights_max: 2
net.inet.udp.recvspace: 786896
```