Cse 190 Quadcopter: Lab 1
Student 1: Zhiye Zhang
Student 2: Xuefei Zhong

For Problem 1:
The result is:
Packet Size(bytes) | Max Single Way Bandwidth:
1 | 3.03KBps
2 | 5.57KBps
3 | 7.72KBps
4 | 9.56KBps
5 | 11.16KBps
6 | 12.54KBps
7 | 13.76KBps
8 | 14.89KBps
9 | 15.89KBps
10 | 16.76KBps
11 | 17.53KBps
12 | 18.32KBps
13 | 18.93KBps
14 | 19.57KBps
15 | 20.14KBps
16 | 20.64KBps
17 | 21.15KBps
18 | 21.60KBps
19 | 21.99KBps
20 | 22.40KBps
21 | 22.73KBps
22 | 23.12KBps
23 | 23.41KBps
24 | 23.70KBps
25 | 23.99KBps
26 | 24.30KBps
27 | 24.54KBps
28 | 24.80KBps
29 | 25.02KBps
30 | 25.24KBps
31 | 25.42KBps
32 | 25.63KBps
33 | 25.81KBps
34 | 25.99KBps
35 | 26.15KBps

36 | 26.32KBps
37 | 26.49KBps
38 | 26.64KBps
39 | 26.76KBps
40 | 26.91KBps
41 | 27.04KBps
42 | 27.20KBps
43 | 27.32KBps
44 | 27.44KBps
45 | 27.54KBps
46 | 27.66KBps
47 | 27.77KBps
48 | 27.86KBps
49 | 27.92KBps
50 | 28.03KBps
51 | 28.12KBps
52 | 28.21KBps
53 | 28.32KBps
54 | 28.40KBps
55 | 28.46KBps
56 | 28.55KBps
57 | 28.63KBps
58 | 28.70KBps
59 | 28.77KBps
60 | 28.85KBps
61 | 28.92KBps
62 | 28.98KBps
63 | 29.04KBps
64 | 29.11KBps
65 | 29.17KBps
66 | 29.23KBps
67 | 29.30KBps
68 | 29.36KBps
69 | 29.42KBps
70 | 29.48KBps
71 | 29.52KBps
72 | 29.54KBps
73 | 29.61KBps
74 | 29.66KBps
75 | 29.72KBps
76 | 29.77KBps
77 | 29.81KBps
78 | 29.86KBps

79 | 29.91KBps
80 | 29.94KBps
81 | 29.98KBps
82 | 30.01KBps
83 | 30.06KBps
84 | 30.05KBps
85 | 30.10KBps
86 | 30.15KBps
87 | 30.18KBps
88 | 30.20KBps
89 | 30.23KBps
90 | 30.27KBps
91 | 30.28KBps
92 | 30.32KBps
93 | 30.35KBps
94 | 30.38KBps
95 | 30.40KBps
96 | 30.43KBps
97 | 30.46KBps
98 | 30.48KBps
99 | 30.51KBps
100 | 30.53KBps
101 | 30.59KBps
102 | 30.62KBps
103 | 30.64KBps
104 | 30.66KBps
105 | 30.68KBps
106 | 30.70KBps
107 | 30.72KBps
108 | 30.75KBps
109 | 30.76KBps
110 | 30.79KBps
111 | 30.81KBps
112 | 30.83KBps
113 | 30.85KBps
114 | 30.86KBps
115 | 30.88KBps
116 | 30.90KBps
117 | 30.92KBps
118 | 30.93KBps
119 | 30.95KBps
120 | 30.96KBps
121 | 30.98KBps

122 | 30.99KBps
123 | 31.02KBps
124 | 31.02KBps

For the first board, we modified the program so that in the setup function, we record the start time and then send a fixed string through the radio by using rfPrint. For the second board, we created another program which just forwards the data when it receives the data from another board. The data is then forwarded back to the first board, and when it receives the data,the program will then output the difference between current time and the start time. The difference represents total time spent on transmitting the data. Since we know the length of the data in bytes, we can just divide the length of data(bytes) by the time, which will give us the bandwidth.

We found that bandwidth maximum bandwidth increases as packet size of rfPrint() increases, but as maximum packet size approach maximum allowed by rfPrint(), the maximum bandwidth tapers off at ~30KBps = 240Kbps.

For Problem 2:

Result:
970.43Bps: .3% lost
1895.99Bps: .3% lost
4850.99Bps: .2% lost
26050.24Bps: 14.7 % lost

When a packet is lost, the data sent will be lost too, so the receiver board will get incomplete data. We changed the rate at which the sender sent the packages by varying the amount of delays between each rfPrint() of 50 byte messages. When receiver shows how many bytes is received before the timeout of 1 second. The number of packets lost is the difference between bytes sent and bytes received.

We observe that when the sender is sending packets at well below the maximum bandwidth found in problem 1, the packet lost rate is generally low (~.3% loss). However, when sender approaches maximum bandwidth, we found that the packet lost drastically increases.

Problem 3:

Maximum dual way bandwidth Result:
26042.78Bps

We found that Maximum bandwidth is slightly lower for dual way communication as opposed to send only. That is most likely caused by processor wasting time polling rfAvailable() and rfRead only support reading single byte from buffer at a time.

Problem 4:

As the boards move farther apart, there will be more packets dropped because of the longer distance between the two boards. The single way transmission time and the throughput are constant for the sender. However, when packets are dropped, the receiver both have received less data and induce additional overhead by having the sender resend data. This lowers the overall bandwidth of the system.

When a packet is lost, the data sent will be lost too. So the receiver board will get incomplete data and the serial monitor will display this incomplete data. In order to check for packet lost, we used two laptops each plugged in with a board, and then open both serial monitors on the laptops. When the serial monitors display any incomplete message, that means the packets have been lost.