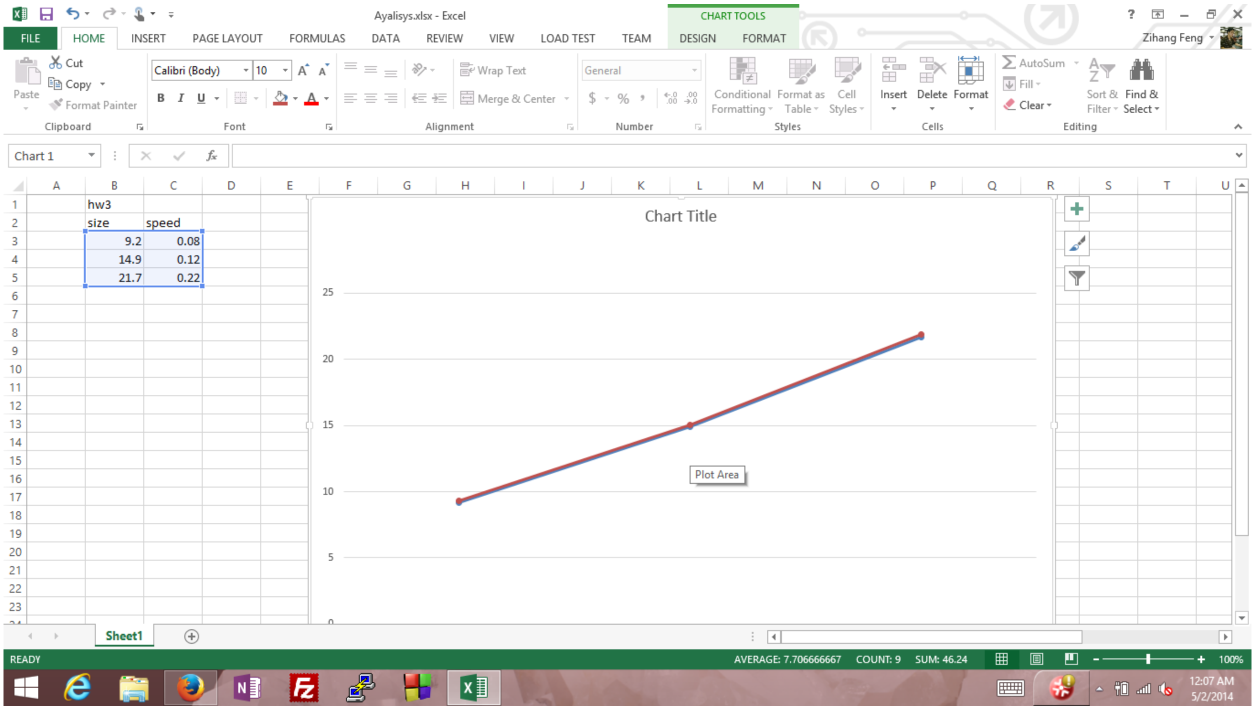
Analysis

Below the two charts from the thrid project and the fourth proejct, respectively. Although both project are somehow similar to each other, they differ by how the client treat the acknoledgement. For the third proejct, when the client receives an acknoledgement, it assumes that number of packs before that have all been acknoledged whether it is out of order of not. Versus, the fourth proejct has the window size to control the number packets. If the acknoledgement is out of order, then the client can't move the winSize unless it receives the expected acknoledgement. On the third proejct, when it encounter out of order of garbled packet, it will send a bulk of packets to the server. On the fourth proeject, if it will only send the required packets.

Here the third proejct seems to be little faster than the fourth project. The reason why it happens is because the fourth proejct has more error checking than the third proejct. Because my fourth project does not create the connect correctly, it only sends to my server no matter what address it is given. If there are a lot of garbled packets, the fourth project should be faster than the third proejct because again the client of the fourth project only sends the required packet back to the server, not a bulk of packets.

The result from the third project



Ayalisys.xlsx - Excel

hw3	
size	speed
9.2	0.08
14.9	0.12
21.7	0.22

READY

AVERAGE: 7.706666667 COUNT: 9 SUM: 46.24

The result from the fourth project

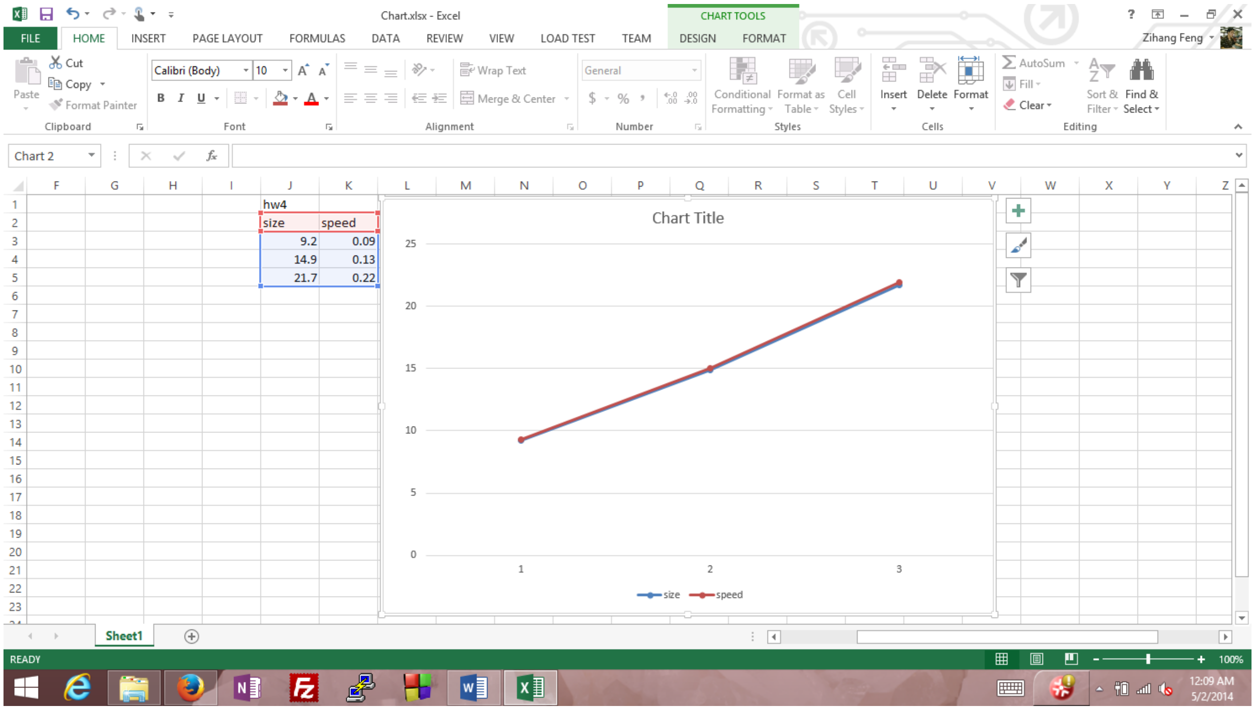


Chart.xlsx - Excel

Chart 2

	J	K
1	hw4	
2	size	speed
3	9.2	0.09
4	14.9	0.13
5	21.7	0.22