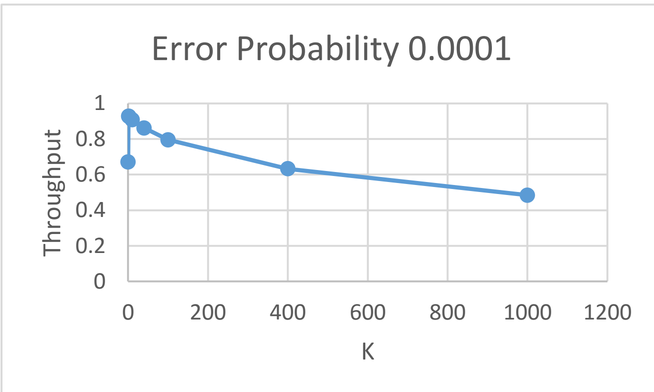
CMPUT 313 ASSIGNMENT 1

A Study of a Combined Error Detection and Error Correction Scheme

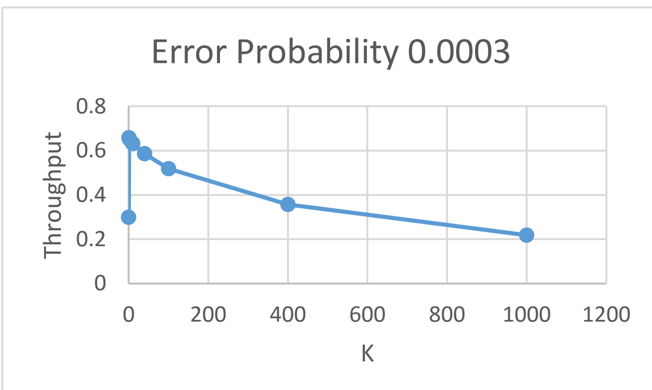
Due Tuesday Feb 2, 5pm

Steven Chan

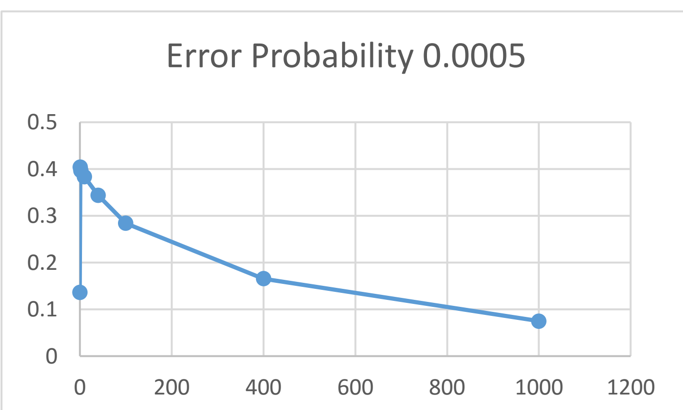
Show a graph of the throughput of the channel versus K for the different values of error probability for the independent error case. Show confidence intervals on your graphs OR provide a table.



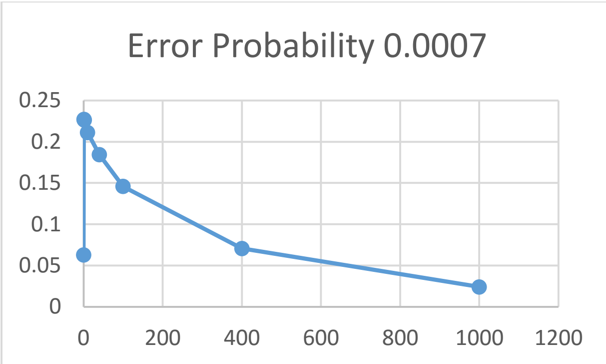
K	Throughput	Error Prob	CI
0	0.671604938	0.0001	(0.6630666478784203, 0.6801432286647896)
1	0.927300282	0.0001	(0.9237250930736154, 0.9308754706500808)
2	0.925043036	0.0001	(0.9248444185537991, 0.9252416542849408)
10	0.90843651	0.0001	0.9080393304088339, 0.9088336900345868)
40	0.86204173	0.0001	(0.8599565488926934, 0.8641269119113216)
100	0.794979414	0.0001	(0.7942846660600588, 0.7956741619369033)
400	0.632768362	0.0001	(0.6322718004335351, 0.6332649227303068)
1000	0.484906603	0.0001	(0.48381531400641803, 0.4859978914675606)



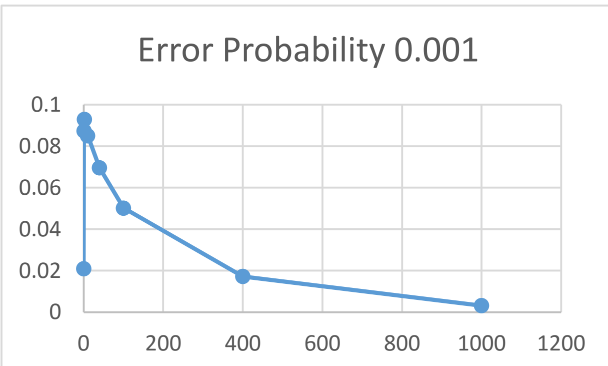
K	Throughput	Error Prob	CI
0	0.299735093	0.0003	(0.2936788634851737, 0.305791321949924)
1	0.658997561	0.0003	(0.6544292637721361, 0.6635658573420642)
2	0.651305811	0.0003	(0.6474327629778636, 0.6551788597351267)
10	0.633298141	0.0003	(1.5253986426450572, 1.5258769319950036)
40	0.586585085	0.0003	(0.5833083716300003, 0.5898617992307018)
100	0.518847184	0.0003	(0.5188471839149379, 0.5188471839149379)
400	0.357422129	0.0003	(0.3538468886251781, 0.3609973691619326)
1000	0.218319848	0.0003	(0.21593885457304252, 0.22070084176099003)



K	Throughput	Error Prob	CI
0	0.136752137	0.0005	(0.13129160219835592, 0.1422126713059176)
1	0.404613943	0.0005	(0.40451463206240657, 0.4047132536617528)
2	0.396607002	0.0005	(0.3907477752630689, 0.4024662293317488)
10	0.383754039	0.0005	(0.3816688450776413, 0.38583923311284396)
40	0.343760935	0.0005	(0.34316516883943515, 0.3443567011304718)
100	0.284286685	0.0005	(0.28101144477302953, 0.2875619253352968)
400	0.165911704	0.0005	(0.16571307995542497, 0.16611032887413354)
1000	0.075117371	0.0005	(0.07313320956370732, 0.07710153222033024)



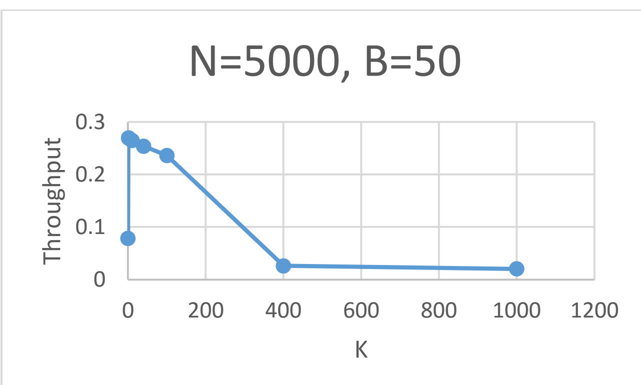
K	Throughput	Error Prob	CI
0	0.062698056	0.0007	(0.05942173494824329, 0.06597437641278034)
1	0.22702538	0.0007	(0.2191798263913786, 0.2348709327397334)
2	0.226381165	0.0007	(0.21297445916103266, 0.23978787101309687)
10	0.210992738	0.0007	(0.21039696790745693, 0.21158850734608625)
40	0.184277616	0.0007	(0.18159666849345346, 0.18695856380311832)
100	0.145820842	0.0007	(0.13966736008724556, 0.15197432356786894)
400	0.070556472	0.0007	(0.07045715994671403, 0.07065578440606832)
1000	0.023973629	0.0007	(0.02198946767977963, 0.02595779033640257)



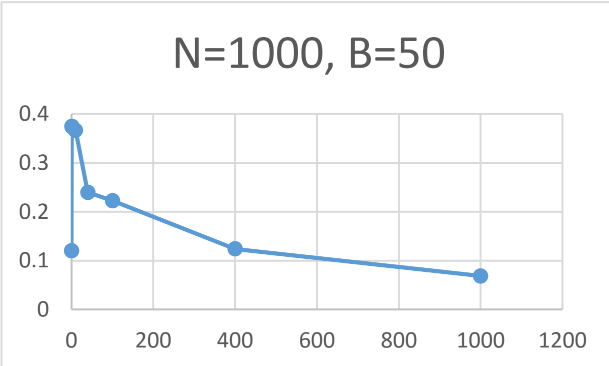
K	Throughput	Error Prob	CI
0	0.020952667	0.001	(0.020853384120269276, 0.021051949013134036)
1	0.087354374	0.001	(0.08646057718123185, 0.08824817157534823)
2	0.092952266	0.001	(0.08540478747974575, 0.10049974526313005)
10	0.085100937	0.001	(0.08490234748358642, 0.08529952729646285)
40	0.069583996	0.001	(0.06362633422581022, 0.07554165713617651)
100	0.05020586	0.001	(0.047823867083456116, 0.05258785294692323)
400	0.017279136	0.001	(0.015491515909009199, 0.019066756177386485)
1000	0.003196484	0.001	(0.0017083628715118771, 0.004684604863979083)

Also show a graph of Throughput versus K for the Independent model case and Burst Model combinations of: (N=5000,B=50), (N=1000,B=50), (N=5000,B=500) and (N=1000,B=500) for e = 0.0005. Show confidence intervals on your graphs OR provide a table.

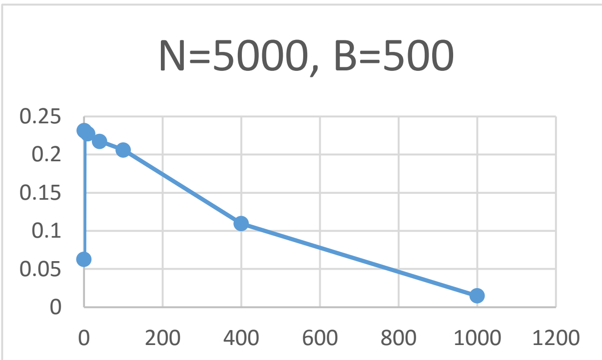
Throughput versus K Independent of e = 0.0005 shown above already.



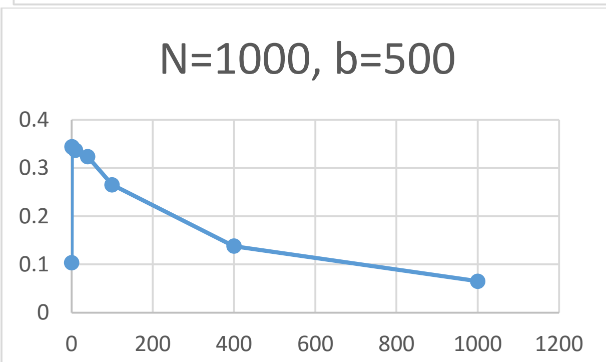
K	Throughput	CI
0	0.078532514	(0.07843323117380054, 0.0786317960666653)
1	0.269582639	(0.2680929768829593, 0.27107230087315326)
2	0.268777638	(0.26778454837268906, 0.26977072702839755)
10	0.264580734	(0.26368707933663094, 0.26547438849457483)
40	0.253861612	(0.25316655132617455, 0.25455667233238394)
100	0.23599952	(0.23341902798346684, 0.23858001266888954)
400	0.025918704	(0.023634522782222382, 0.028202885347371136)
1000	0.020297673	(0.018015887032625605, 0.02257945808774199)



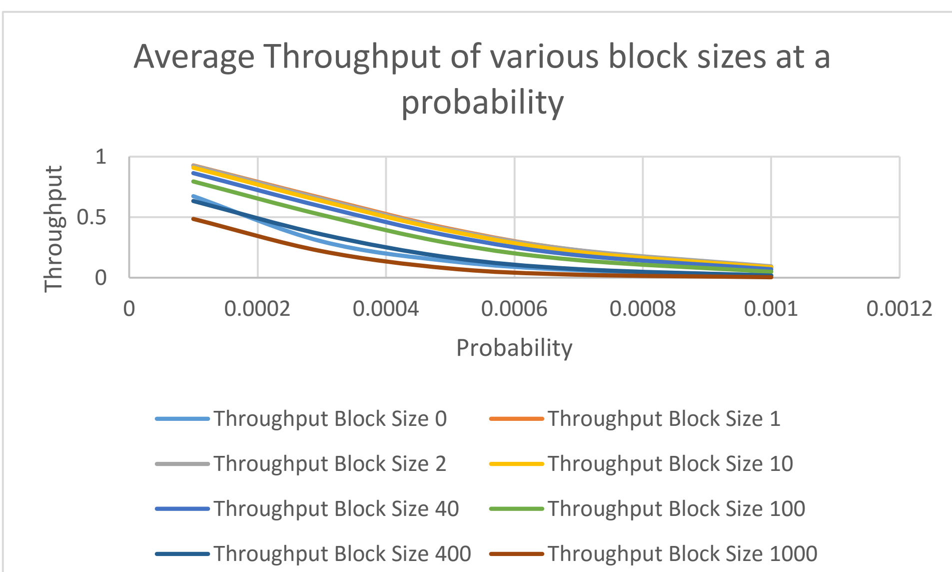
K	Throughput	CI
0	0.120117959	(0.11555096617856049, 0.12468495125033939)
1	0.375015849	(0.37263238978717056, 0.3773993081714809)
2	0.373568919	(0.37108619574629814, 0.37605164238556926)
10	0.367117766	(0.3666212908544055, 0.36761424038659657)
40	0.239624886	(0.23138345459648646, 0.24786631795582653)
100	0.222568653	(0.2169114201028781, 0.2282258865286125)
400	0.123833808	(0.11648470331347566, 0.13118291330569337)
1000	0.068404755	(0.060666525589338535, 0.07614298395016801)



K	Throughput	CI
0	0.062538112	(0.06144600475016251, 0.06363021857167485)
1	0.231345101	(0.22975612858070132, 0.23293407417024153)
2	0.230700806	(0.22901255383561417, 0.2323890575503185)
10	0.227309083	(0.22671331304614586, 0.22790485248477518)
40	0.21739	(0.2155034075057897, 0.21927659309407235)
100	0.206099852	(0.2055043538676382, 0.206695350333505)
400	0.109434528	(0.10933521604390918, 0.10953384050326347)
1000	0.01486365	(0.014069985453691893, 0.01565731451634107)



K	Throughput	CI
0	0.103323837	(0.09875684412128054, 0.10789082919305944)
1	0.343817858	(0.338951628945953, 0.34868408731391987)
2	0.342691488	(0.33752742356335075, 0.3478555525730347)
10	0.337044502	(0.3313846896980557, 0.34270431436503423)
40	0.323285644	(0.31524280133731336, 0.33132848726630776)
100	0.265259623	(0.253448908502871, 0.27707033840922873)
400	0.137913104	(0.12927294036287099, 0.14655326832669455)
1000	0.065208271	(0.061438364378215995, 0.06897817742579958)



Assumptions in study:

- If K>0, K blocks of size (F/K + r)
- F is divisible by K
- If K=1, only 1 block and Hamming applied to entire frame
- If K=0, no error correction. Frame of size F transmitted
- 50 bit time units for acknowledgement of transmit
- Error detection is 'perfect'. Any existing bit errors are always detected
- Independent errors follow a random number (0,1) and is in error if it is <= error_prob
- Burst errors only occur during burst periods, not during non-burst. Errors follow probability e'. e'=e*(N+B)/B
- Simulation therefore:
    - Actual message bits are not generated.
    - Details of HSBC and error decoding schemes not implemented
    - Ignores time required for error checking, splitting message into blocks, etc
    - Only looks at time to transmit encoded blocks and receive back acknowledgement

Input format is as follows:

- Python a1.py M A K F E B N R T t1 t2 t3 tt
- python a1.py I 50 0 4000 0.0007 0 0 5000000 5 32 60 12 38 22

Where M = I or B, A = 50 bit time units (default), F = 4000 bits of frame size (default), R = 5000000 bit time units for length of simulation (default), T = 5 trials (default)

Observations:

Ob1: Increasing values of K results in a decrease in throughput output rate.

Ob2: The higher the error probability, the steeper the crash in throughput.

- With a lower error probability and less blocks, a chance of error occurring was less likely and as such, fewer frames are required to be retransmitted for failure.
- However, as K increases and progresses at a higher error probability, there is a large drop in throughput. This is due to there still being big enough blocks and multiple bit errors in the same block occurring causing retransmission or failure is likely. This is a result of the HSBC check bits failing to correct bits and as a result retransmission is inevitable.

Ob3: After the spike drop in throughput, the throughput mellows down and does not continue to drop as severely. Appears to balance out.

- This is due to the numerous amounts of K at this point. The frame has been split up into so many blocks that these small blocks have a small chance of getting more than 1 error.

As a result, throughput does not drop that dramatically as there are less retransmissions.

Ob4: For all cases of K various block sizes, the throughput is highest when the error probability is lower. Conversely, the throughput is low when the error probability is high.

- Naturally, with a low error rate, it makes sense that throughput would be high since more frames are transmitted without failure.

Ob5: As K increases, the more accurate the average throughput becomes

- After the initial throughput high and the drop, it was observed that the throughput slowed down and stayed at roughly the same rate compared to K values near the beginning.
    - (Data drawn from Error probability 0.0007, Independent Model, Throughput vs K graph above)
    - Example variance: K=0 Throughput= 0.062698056 | K=1 Throughput= 0.22702538
        - Difference in throughput=0.22702538-0.062698056=0.164327324
    - Example variance: K=400 Throughput= 0.070556472 | K=1000 Throughput=0.023973629
        - Difference in throughput=0.070556472-0.023973629=0.046582843
    - 0.164327324/0.046582843=3.52763621576 times in difference
    - Despite K only increasing one, the variance is huge whereas K increases by 600 later on and yet does not vary by much.

Ob6: Highest and lowest rates of Throughput

- Highest: 0.927300282 @K=1,error_prob=0.0001
- Lowest: 0.003196484 @k=1000,error_prob 0.001
- This matches the observations and conclusions discussed earlier.

Ob7: struggling K=0 throughput in every scenario

- When K is 0, it is interpreted from the previous assumptions that there will be no error correction used and the frame of size F is simply transmitted. Since HSBC was not used, there were no check bits and as a result, failures were inevitable and retransmissions were occurring. As the error probability increases, the throughput at K=0 continues to drag behind as the chances of getting a bit error increases more and more, which results in more and more retransmissions.

Ob8: Spiking K=1 throughput in every scenario

- When K is 1, HSBC is in full effect so there are check bits. Single bit errors can now be corrected, which results in a great decrease in retransmissions and this is shown in all the graphs.

Ob9: Confidence Intervals

- Initially the confidence interval is quite wide. However, as K increases, the confidence interval becomes tighter and tighter. This supports the previous conclusion that throughput stabilizes at a high K value.

Ob10: HSBC or no HSBC?

- From the data, it can be seen that K=0 (no HSBC) equalizes throughput around K=400 to K=1000, which is also generally the period where throughput stabilizes.
- As a result, using HSBC scheme produces greater throughput than without.

Conclusion:

In conclusion, using the Hamming Single-Bit Error Correction scheme overall increases throughput rates as opposed to without. This is shown by all the data and graphs where the throughput rate with HSBC is significantly higher and the drop off rate equalizes at roughly the rate of throughput without HSBC anyway.