CMPUT 313 Assignment 1

A Study of Combined Error Detection and Error Correction Scheme

Andrew Charles and Jake Brand

Consulted with: Kirby Banman, Anthony Ou, David Liao

Once our simulator was up and working we started running a series of tests with varying values of K blocks, over a spread of e error probabilities from 0 to 1. Our intention was to identify how the number of blocks, and the overhead of the introduced check bits, would affect the throughput and the average frames correctly received. Before we delve into the analysis of the results, here are a few of the assumptions we made:

1) As our simulation was run for a sufficiently long enough time, we safely included frames that finished getting received (slightly) after the R time units had been used up. See the discussion forum's "Stats Clarification" if this is unclear.
2) Zero time units are required for splitting the F bits into K blocks, calculating the number and values of r check bits, and inserting the r check bits into the blocks.
3) We trimmed the error probabilities to a subsection of the 0-100% range in order to clearly illustrate the effects of changing e over the K cases.
4) The provided inputs are valid for the scope of this assignment. Although we recognize that the provided values may not produce the most clear results, we expect that they are the results you are most familiar with. As such we used the following values to generate our graphs:

    A = 400,   F = 4000,   R = 400000,   T = 5

    Seeds = [1534546 2133323 377 456548 59998]

Throughput Similarities:

All cases with K>0 follow roughly the same general trend. The throughput is highest with the lowest chance of error. This is because fewer errors allow more frames to be transmitted successfully. The throughputs then decreases slowly as e grows, because more frames are being retransmitted, until the throughput gets to a pivot point at an error probability of about 0.001. Following this pivot point the throughput decreases more rapidly because the HSBC check bits are no longer able to avoid retransmitting frames as efficiently. At a point just before the throughput reaches zero, the rate of decline becomes less steep once more as the number of frames being received gets closer to zero. The last point of interest is that all cases eventually converge at a point where no frames are being received correctly and the throughput reaches zero. Theoretically this point would be guaranteed at e=1, but as you can observe the throughput reaches essentially zero well before this error probability.

As the values of K increased, the rate of throughput decline over the e probability values was reduced. This produced a more shallow line for low values of e, and was due to the fact that fewer frames had to be retransmitted with each increase in K. As the number of blocks grew, another phenomenon occurred in which the variance would start out very small but increase as the throughput decreased more rapidly. Conversely, the highest throughput values became lower for quite high values of K. This can be explained as the near-ideal throughput scenario. As the frame was divided into more and more blocks, there was a lower chance of multiple bit errors occurring, so fewer frames were retransmitted each trial and the variance was kept low. Due to the overhead of the HSBC however, the maximum throughput value shrunk as K grew. As the graph illustrates, the higher the K value the longer the throughput stayed close to this near-max value and the greater guarantee of the value being close to the measured average.

The best case for K at low values of e was 40. The best case for K at medium values of e was 80. The best case for K at high values of e was 400.

Throughput Case-By-Case Analysis:
K=0:
In the special case of K=0, the complete lack of HSBC check bits ensured that many frames had to be resent and the throughput started quite low. As the probability of a bit error increased the number of frames having to be resent increased as well and the throughput dropped until it leveled out at essentially 0 quite early on.

K=1 and K=2:
Unlike the previous case, the HSBC check bits are included in the bits being sent. This means that the time required to send one frame is going to be greater than the K=0 case. As single bit errors can now be corrected, the number of frames that have to be re-sent is going to be much lower, especially at small values of e where multiple bit errors are unlikely. As the confidence intervals between these cases overlap, we can not state that the two are statistically significant for any values of e. As they have no other notable differences they are discussed in the same way.

K=5:
The throughput at this number of blocks is higher on average than the K=1 case, but is often not statistically different than the K=2 case. One notable difference is that the confidence interval starts off quite small then quickly grows before diminishing as it approaches its end case. This is the first obviously observable occurrence of the near-ideal phenomenon discussed above. As K is still relatively small, the throughput does not stay close to this near-ideal scenario for very long before it starts to decrease with the retransmission of more frames.

K=50:
50 blocks is a much nicer balance between check bit overhead and lack of frame retransmissions. The confidence intervals follow the same growth and diminish patterns as was evident in the K=5 case, but over a greater set of e values, as the same phenomenon occurs.

K=400:
Here we can easily see the near-ideal phenomenon. Breaking the frame into 400 blocks approaches the ideal condition where the HSBC is able to correct almost all of the errors. As the check bit overhead is very high with this many blocks, the throughput's near-max value is statistically lower than the K=5 and K=50 case but remains close to this value until the pivot point. Before this pivot point, the variance stays at a much lower level but as the value of e grows past this point and frames are eventually retransmitted, the throughput depreciates at very fast rate while the variance increases.

K=4000:
Similar to the previous cases, splitting the frame into a very large number of blocks resulted in low variance results that stayed essentially constant for a long range of error probabilities. The huge overhead of 4000 blocks brought the average of this case to a much lower value than all previous HSBC trials.

Throughput conclusion:
As expected, the throughput lost by using the HSBC bits is much lower than the throughput gained by correcting one bit errors. It is evident that, at these input values, implementing HSBC will produce a better throughput. At low values of K, you can see that increasing the value of K will actually produce better throughput values up to a certain value of K. Given these inputs, the optimal value of K at a low probability of error is 40. At this value the HSBC is able to correct enough bits to keep the throughput high while requiring few enough check bits to keep the overhead low.

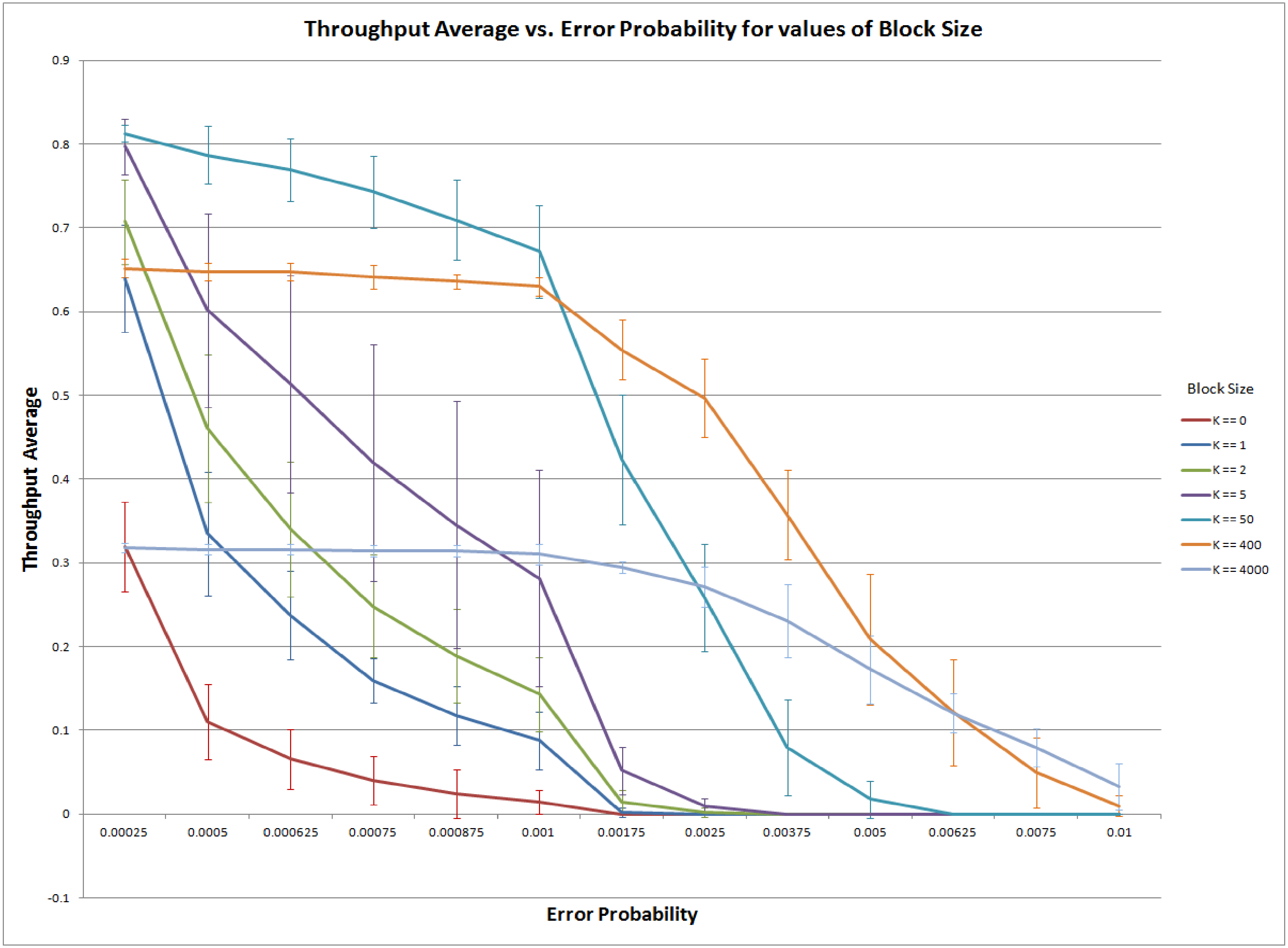
If we could guarantee the probability of an error occurring were in a range somewhere around 0.001, the most desirable K value to use would be around 80 as this produces a throughput that is quite high on average, although K value of 100 with a slightly lower throughput has a much lower variance. As the graph indicates, there is a value e where the previously optimal K value is no longer desirable. In our graph you can see that although K=50 is significantly better with low error probabilities, given higher values of e (anything after 0.00175) the K=400 option is significantly better. Furthermore, as the error probability becomes very large, (0.01 for example) our tests determined that a K value equal to the number of bits in the frame (F) generated the best average throughput. At this very high probability of error, only the smallest sized blocks were transmitted successfully. That said, the results of very high K values had large enough variance that the confidence intervals overlapped. These overlapping variables ensure we can not call K=F the statistically significant ‘best’ option.

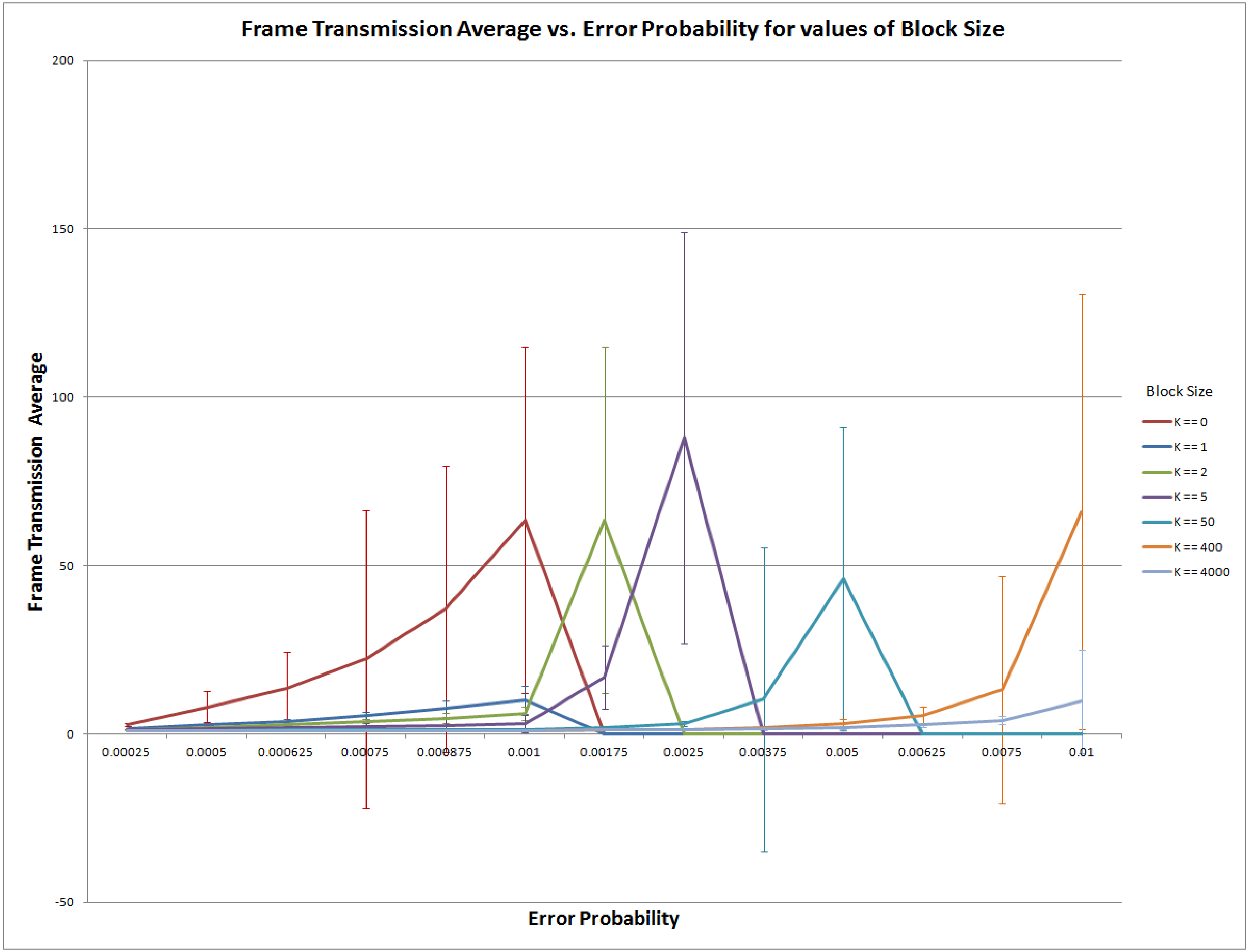
Higher values of K have a higher staying power, where the throughput is affected less by an increase in e, this results in intersection points between values 1, 50, 400, and 4000. Despite having a higher throughput during higher error probabilities, the overall throughput is lower than smaller values of K.

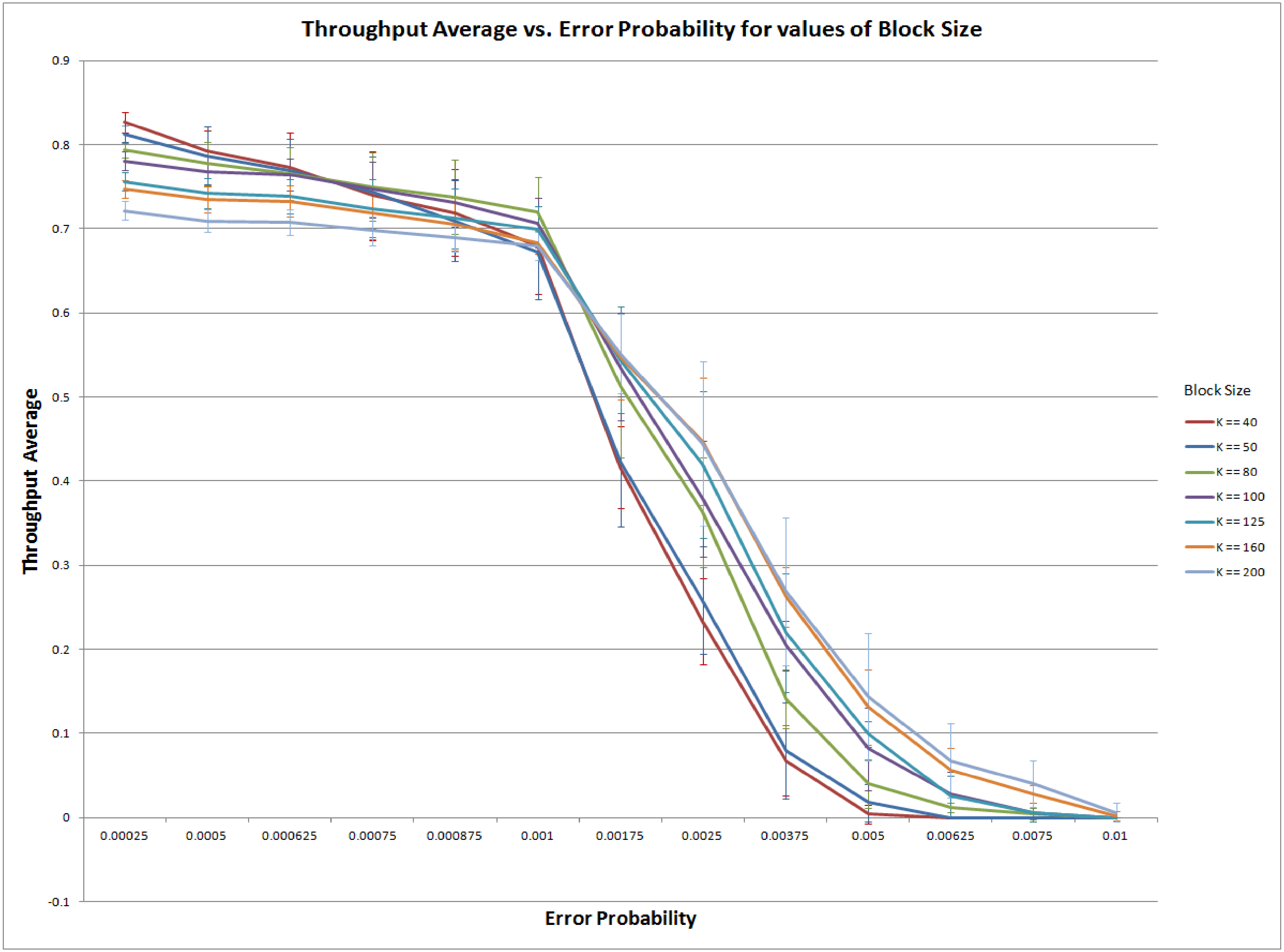
A note on the ACK:
We chose the value for A that we used simply because we wanted an ACK that was quite small compared to the time required to transmit a frame. Had the size of a frame been smaller, or inversely the size of the ACK been larger, we would expect the throughput of our graphs to be slightly lower than they currently are. The rate of throughput decline over the e probabilities and the variance expressed by the confidence intervals would not be expected to change very much. The exact opposite can be said about decreasing the size of the ACK relative to the size of the frame. It is worth noting that at some very large value of A, so few frames could be transmitted within the R time units that every simulation would result in the same throughput of 1 or 0 frames per trial. We believe that as A approaches this extreme value the throughput values at all e probabilities would diminish significantly. As this extreme would produce results that do not aid in answering the questions of this assignment, we did not run this simulation and we did not plot those graphs.

The zeroing out points on the frame transmission average graph match up with the points on the throughput graph. On this graph however, the frame transmission average grows until the zeroing out point, whereas the throughput graph decreases to the zeroing out point. This asymptotic behaviour could be represented better if the frame transmissions averages with values of zero were removed. There is a much higher variance in values in the frame transmission average graph that grows greatly as the frame transmission average grows. This graph does not seem to be plotting very useful data. For K values of 400 and 4000 both frame transmission averages require a very high error probability before increasinging.

In our initial testing we found an asymptote between 0.00175 - 0.0025 for K = 1 and 0.0025 - 0.00375 for K = 2. The value grew from 60/80 - 445/445 and then dropped to 0 on the next error probability. We believe this is an edge case due to the way the formula is derived (averaged (Total / Correct)) where 1 correct frame and 445 incorrect frames occurred.


Throughput Average vs. Error Probability for values of Block Size
Throughput Average
Error Probability
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0.00025
0.0005
0.000625
0.00075
0.000875
0.001
0.00175
0.0025
0.00375
0.005
0.00625
0.0075
0.01
Block Size
K == 0
K == 1
K == 2
K == 5
K == 50
K == 400
K == 4000


Frame Transmission Average vs. Error Probability for values of Block Size
Frame Transmission Average
200
150
100
50
0
-50
0.00025
0.0005
0.000625
0.00075
0.000875
0.001
0.00175
0.0025
0.00375
0.005
0.00625
0.0075
0.01
Error Probability
Block Size
K == 0
K == 1
K == 2
K == 5
K == 50
K == 400
K == 4000


Throughput Average vs. Error Probability for values of Block Size
Throughput Average
Error Probability
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
0.00025
0.0005
0.000625
0.00075
0.000875
0.001
0.00175
0.0025
0.00375
0.005
0.00625
0.0075
0.01
Block Size
K == 40
K == 50
K == 80
K == 100
K == 125
K == 160
K == 200