A Study of a Combined Error Detection and Error Correction Scheme

CMPUT 313 - LAB 1

mccarten@ualberta.ca
wanxin@ualberta.ca

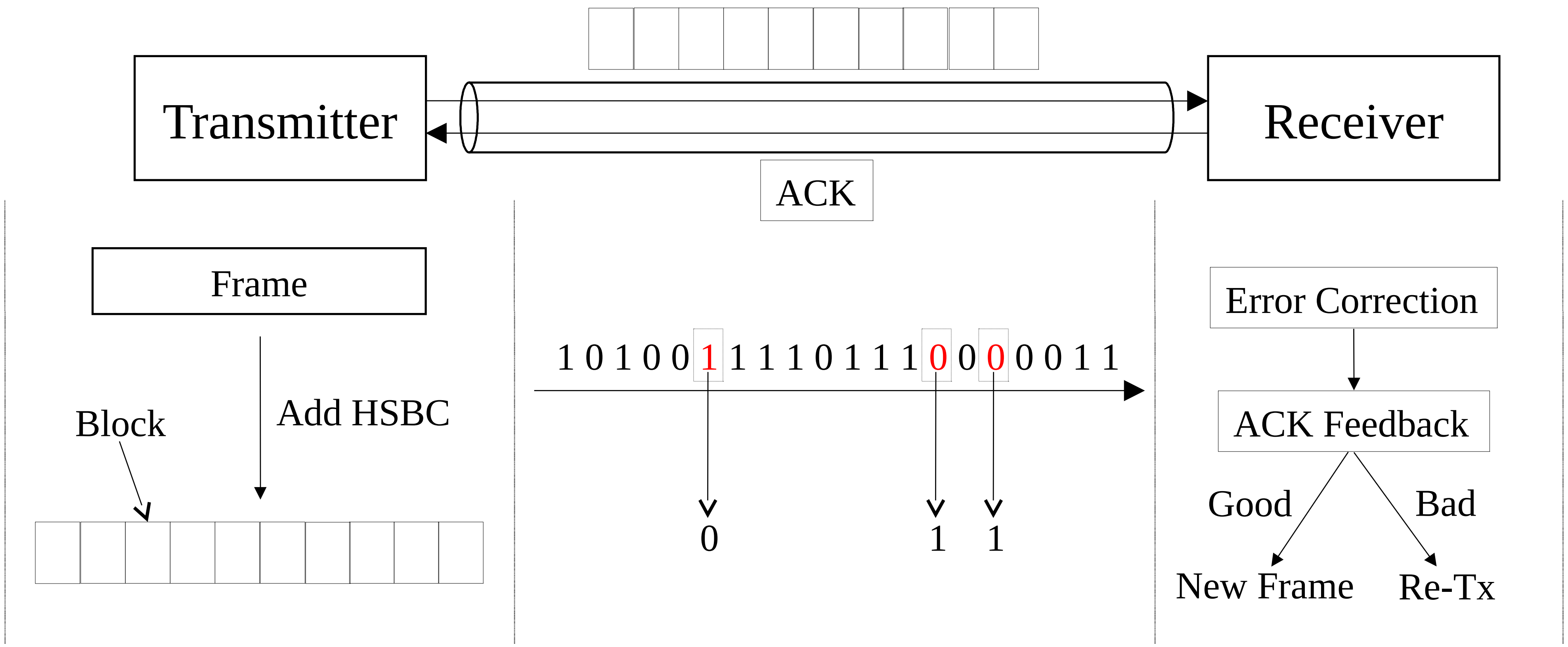
Overview

- Write a simple simulator to investigate the impact of *error-correction* encoding on the ***throughput*** of a communication channel.

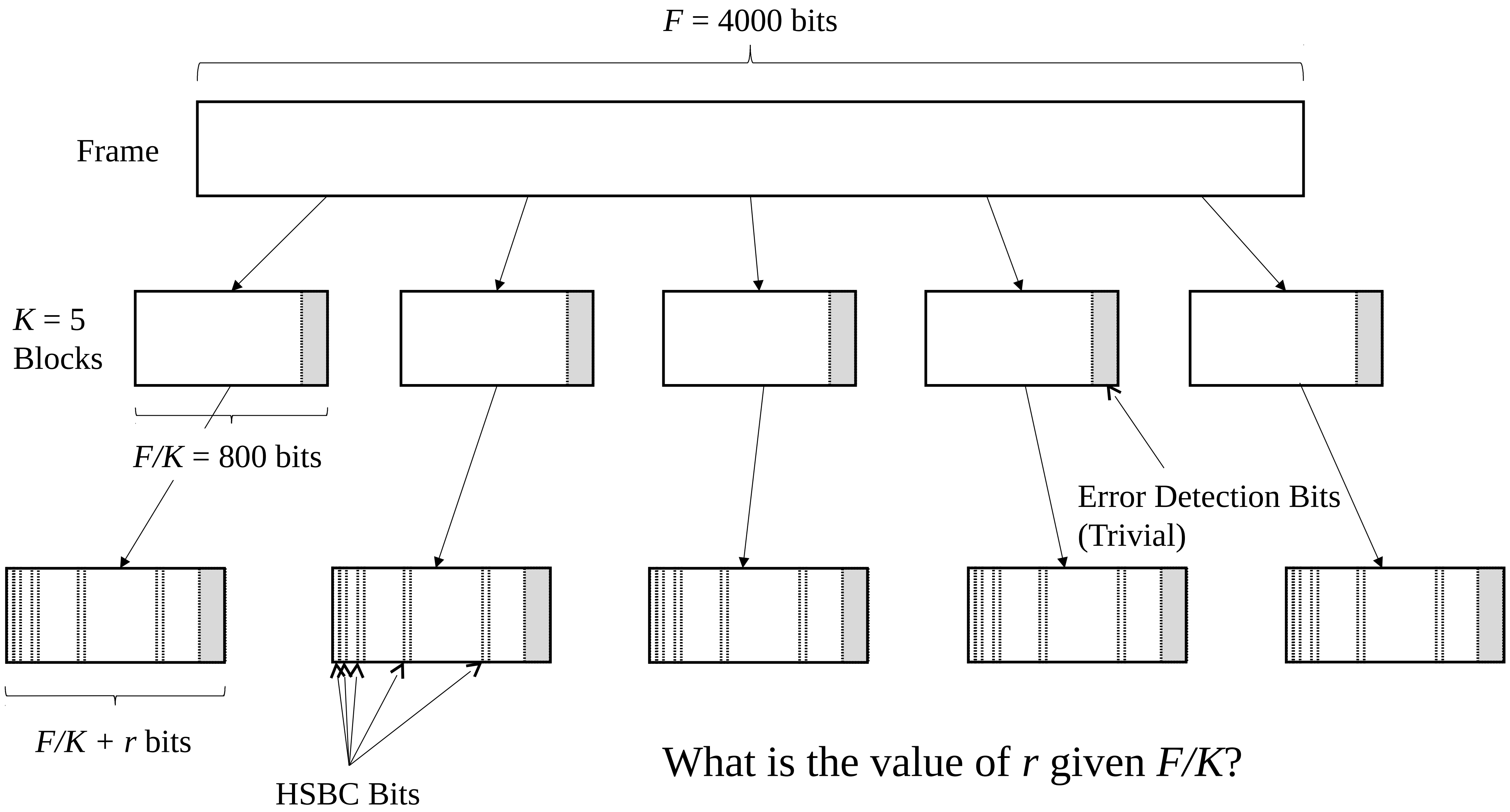
- Simulate passing bits over an *error-prone* channel.

- Caveats:
  - the actual message bits need NOT be generated;
  - the details of Hamming's Single-Bit Error Correction (HSBC) and error decoding schemes need NOT be implemented.

System



Transmitter
Receiver
ACK
Frame
Add HSBC
Block
1 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 0 0 1 1
0
1
1
Error Correction
ACK Feedback
Good
Bad
New Frame
Re-Tx



What is the value of r given F/K?

Value of r

$$F/K = 1 \rightarrow r = 2$$

3	2	1
d 1	c 2	c 1

$$F/K = 2, 3, 4 \rightarrow r = 3$$

7	6	5	4	3	2	1
d 4	d 3	d 2	c 3	d 1	c 2	c 1

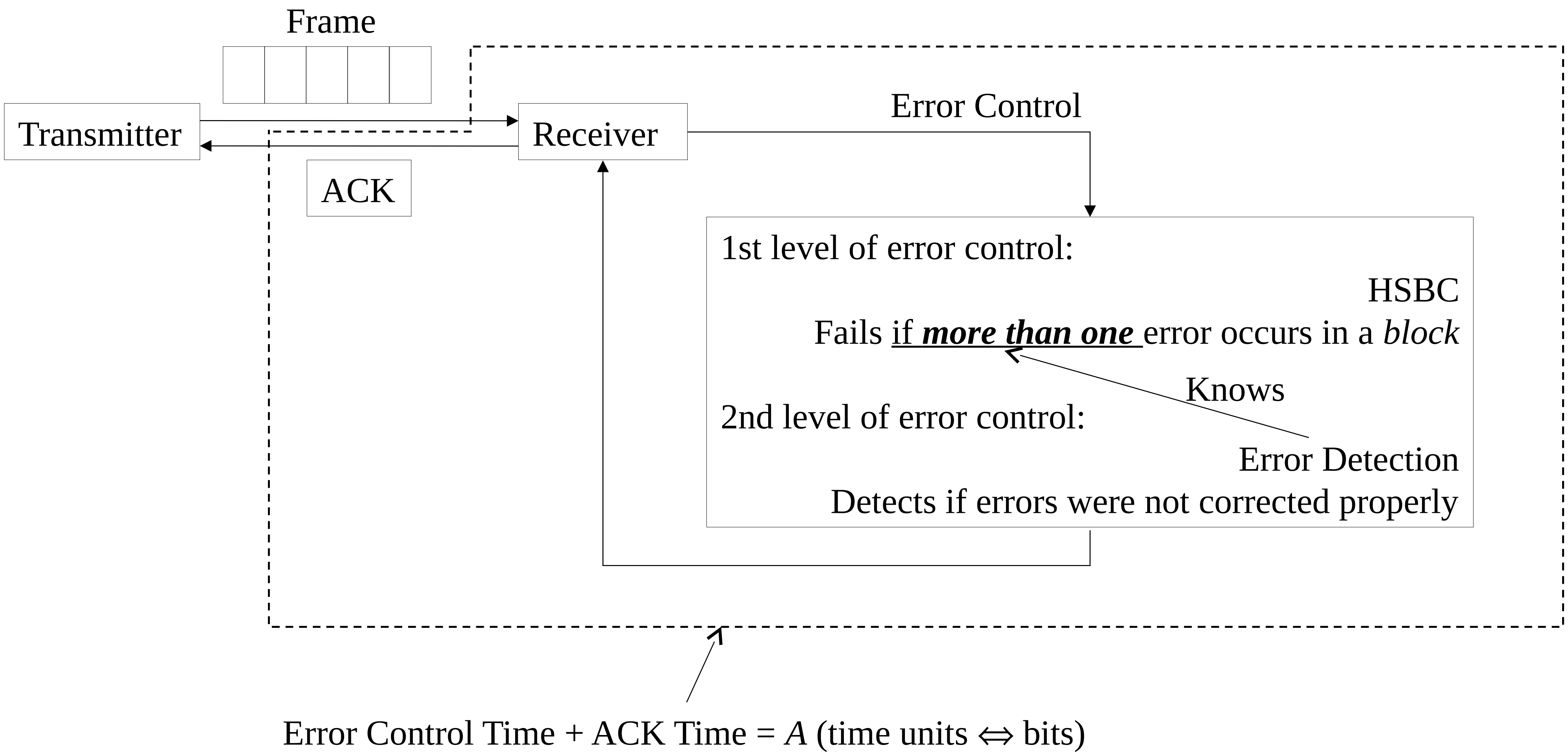
$$F/K = 5 \rightarrow r = 4$$

9	8	7	6	5	4	3	2	1
d 5	c 4	d 4	d 3	d 2	c 3	d 1	c 2	c 1

Generating Bits in Error

- Independent errors:
  - generate a random number x in (0, 1) for each bit;
  - if $x \leq e$, then that bit is in error.

- Burst errors:
  - generate a random number x in (0, 1) for each bit in the *burst* periods;
  - if $x \leq e' = e \times \frac{N+B}{B}$, then that bit is in error.


Frame
Transmitter
Receiver
ACK
Error Control
1st level of error control:
HSBC
Fails if ***more than one*** error occurs in a *block*
Knows
2nd level of error control:
Error Detection
Detects if errors were not corrected properly
Error Control Time + ACK Time = *A* (time units ⇔ bits)

Code Example

```
While (clock < MAX_SIM_TIME) {
   ……
   int frame_ok_count = 0;
   CalcLengthOfBlocks4Tx(); // Transmitter
   for (each block i) GenerateRandomErrors(); // Channel
   for (each block i) // Receiver
      if (NumOfErrors(i) > 1) re_tx = true;
   if (re_tx == false)
      ++frame_ok_count;
   UpdateClock(clock);
   ……
}
```

Key Terms

- The average number of frame transmissions:

$$\frac{\text{the total number of frame transmissions including retransmissions}}{\text{the number of frames correctly received}}$$

- Throughput:

$$\frac{F \times \text{the total number of correctly received frames}}{\text{the total time required to correctly receive these frames}}$$

Key Terms

- Confidence intervals:

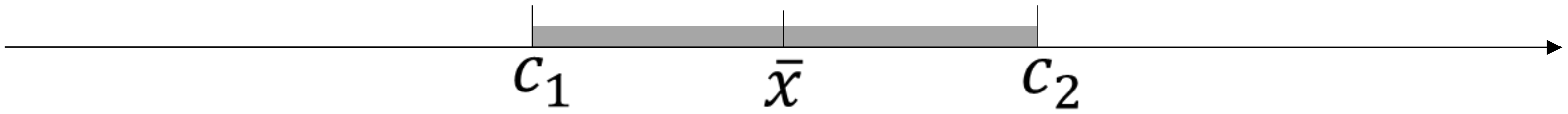
  Suppose we have run T trials of simulation, and got T values, x_1, x_2, x_3, …, x_T. Then their mean value is

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_T}{T}$$

  But the above $\bar{x}$ may not be the "***true***" mean, unless we let $T = \infty$!

Confidence Intervals

- Intuition: the “true” mean is probably somewhere near $\bar{x}$.



- A confidence interval is a range (c_1, c_2) which includes the true mean with a certain probability, say, 95%.

- To calculate (c_1, c_2), we also need the standard deviation of the T values.

Confidence Intervals

- We can calculate the standard deviation as

$$s = \sqrt{\frac{\sum_{i=1}^{T}(x_i - \bar{x})^2}{T-1}}$$

- Then for the 95% confidence interval, where $\alpha = 0.95$,

$$c_1 = \bar{x} - t_{\left[1-\frac{\alpha}{2};T-1\right]}\frac{s}{\sqrt{T}}, c_2 = \bar{x} + t_{\left[1-\frac{\alpha}{2};T-1\right]}\frac{s}{\sqrt{T}}$$

Confidence Intervals

- Here $t_{\left[1-\frac{\alpha}{2};T-1\right]}$ is the ***t*-distribution**, with parameters $1-\frac{\alpha}{2}=0.975$ and $T-1=4$.

- $t_{[0.975;4]}=2.776$.

- Then all we need for the 95% confidence interval is:

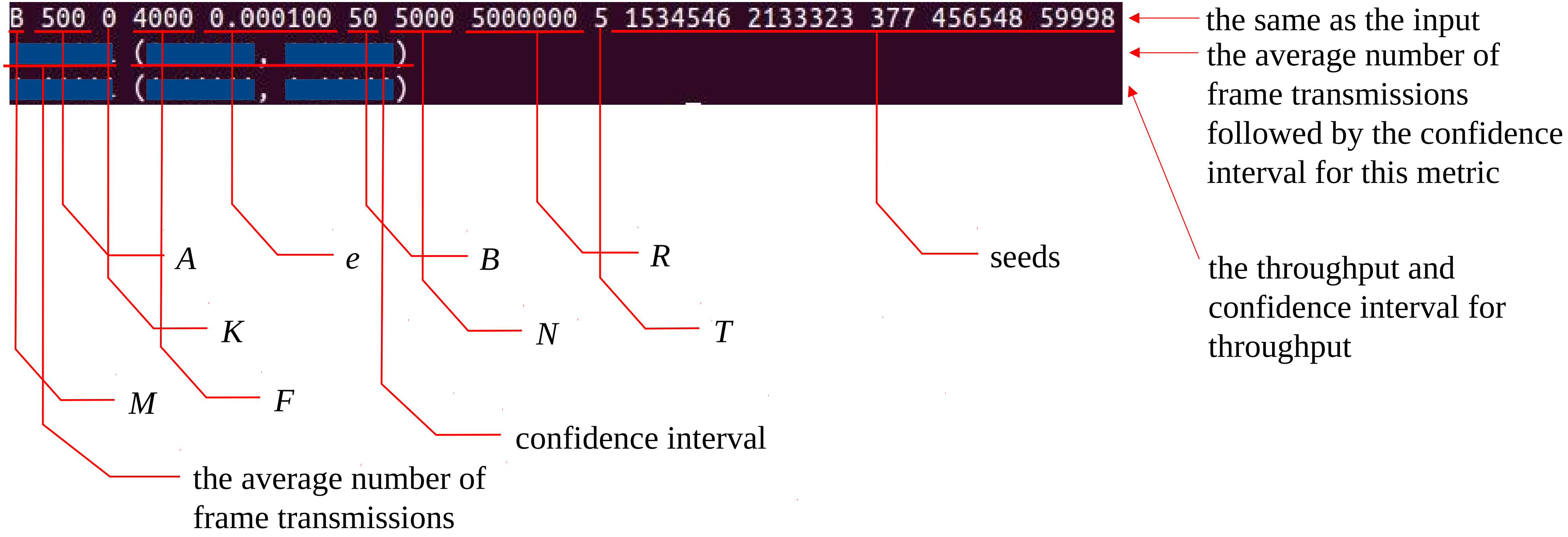
$$\boxed{c_1=\bar{x}-2.776\frac{s}{\sqrt{T}},c_2=\bar{x}+2.776\frac{s}{\sqrt{T}}}$$

Inputs

- Command line arguments:
  - ***M*** (character): the error model used: "I" for Independent, "B" for Burst.
  - ***A*** (integer): the feedback time, say, 50 bit time units.
  - ***K*** (integer): the number of blocks. Choose *K* such that *F* is a multiple of *K*.
    $K = 0, 1, 2, 10, 40, 100, 400, 1000$.
  - ***F*** (integer): the size of a frame, say, 4000 bits.
  - ***e*** (floating): the probability of a bit in error.
    $e = 0.0001, 0.0003, 0.0005, 0.0007, 0.001$.
  - ***B, N*** (only for Burst error model): $B = 50, 500$, and $N = 5000, 1000$.
  - ***R*** (integer): the length of the simulation (in bit time units), say, 5,000,000 bit time units. *R* should be long enough for stable results.
  - $T\ t_1\ t_2\ t_3 \ldots t_T$ (integer): the number of trials (say, 5), followed by seeds for the trials.

Outputs

- An instance of the output is as follows:



Deliverables

- Codes, and a write-up, no more than 6 pages, typically including:
  - graphs that show
    - the throughput versus *K* with different *e* for independent error and
    - the throughput versus *K* with (*N*=5000, *B*=50), (*N*=1000,*B*=50), (*N*=5000, *B*=500), (*N*=1000, *B*=500), e=0.0005, for independent and burst error,
  - tables for corresponding confidence intervals (or drawn on the graphs), AND
  - *in-depth discussion of results* and conclusions.

- Languages: C/C++, Python, Java. (Personally, C/C++ is preferred.)

Tips

- Read specifications very carefully.
- Test your assignments on the lab machines before you submit.
- Remember to average the results over T trial runs.
- Make readable codes and write-ups.