CS 403/534 – Distributed Systems
Homework #1

Assigned: 2/04/2019
Due: 7/04/2019 by 11:55 (sharp)

1. (**10 pts**) Consider a p2p network, where the node ids are randomly selected 32-bit unsigned integers. Supposing that there are about 2^{17} peers in the network, what is the probability that we have at least two peers having the same id? Are 32-bit numbers sufficiently large for a network of that size?

   (**Hint**: use birthday paradox and the approximate formula $p(n, d) = 1 - e^{\frac{-n^2}{2d}}$)

   $p(2^{17}, 2^{32}) = 1 - e^{\frac{-(2)^{34}}{(2)^{33}}} = 1 - e^{-2} \cong 0.86$

   Probability that we have at least two peers having the same id is too high. 32-bit unsigned integers are not sufficiently large for a network of that size.

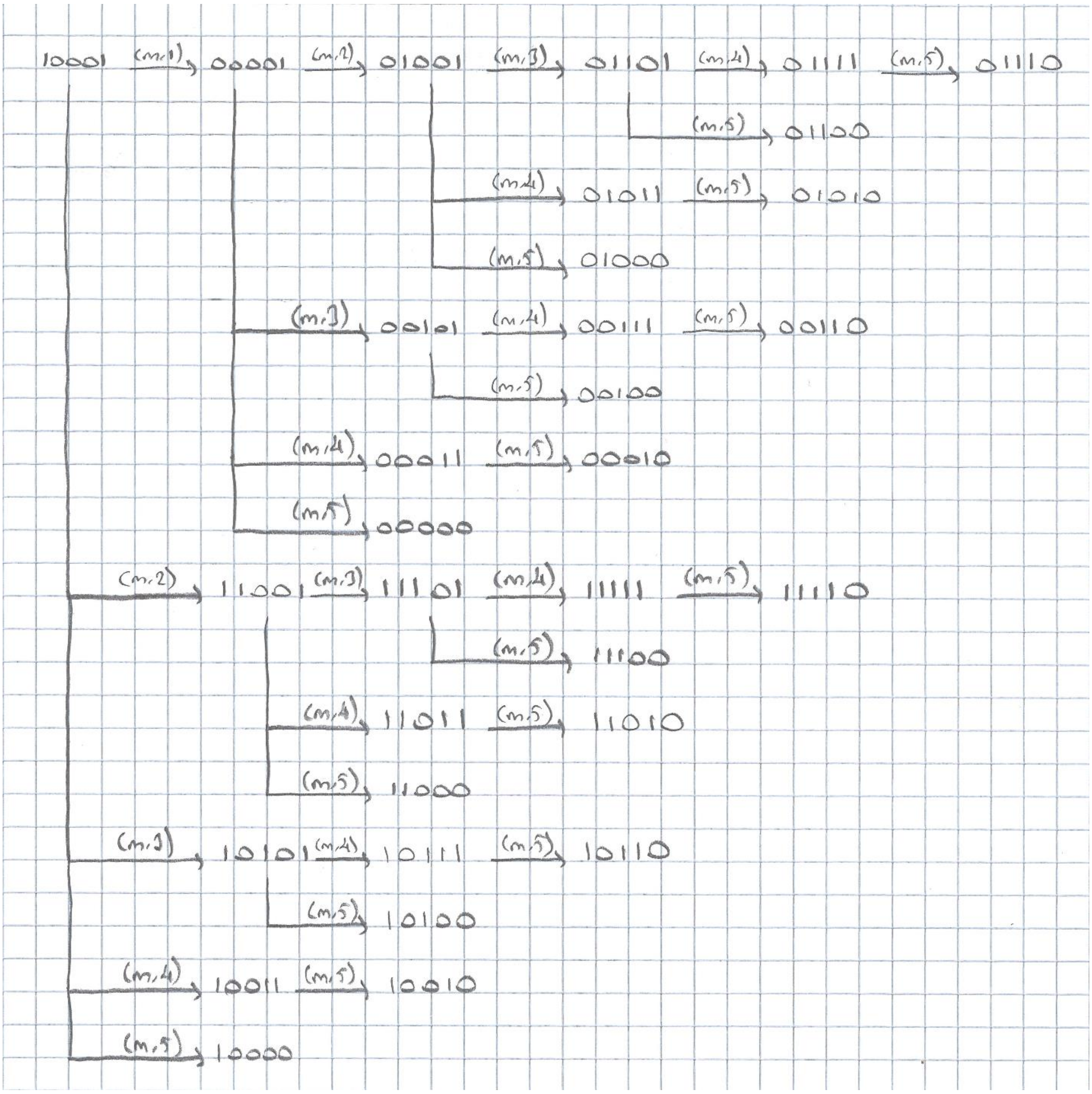
2. (**10 pts**) We want to migrate a virtual machine whose memory image is 800 Gbit on a network with 100 Gbit/s bandwidth to another computer. Approximately, 2% of the memory image changes every second. We do not want to use more than 50% of the bandwidth for the migration operation and maximum allowed down time is 10 ms. Sketch a strategy for the migration operation. Give the total time needed for the migration process.

1st Step:	(800 Gbit) / (50 Gbit/s) = 16 s	
2nd Step:	(800 Gbit) * (0.02) * 16 = 256 Gbit (Changed memory image amount during the first step)	
	⇒ (256 Gbit) / (50 Gbit/s) = 5.12 s (Repeat the computation in the first step with new values)	
3rd Step:	(800 Gbit) * (0.02) * (5.12) = 81.92 Gbit	
	⇒ (81.92 Gbit) / (50 Gbit/s) = 1.6384 s	
4th Step:	(800 Gbit) * (0.02) * (1.6384) = 26.2144 Gbit	
	⇒ (26.2144 Gbit) / (50 Gbit/s) = 0.524288 s	
5th Step:	(800 Gbit) * (0.02) * (0.524288) = 8.388608 Gbit	
	⇒ (8.388608 Gbit) / (50 Gbit/s) = 0.16777216 s	
6th Step:	(800 Gbit) * (0.02) * (0.16777216) = 2.68435456 Gbit	
	⇒ (2.68435456 Gbit) / (50 Gbit/s) = 0.05368709 s	(Still > 0.01 s = 10 ms)
7th Step:	(800 Gbit) * (0.02) * (0.05368709) = 0.85899346 Gbit	
	⇒ (0.85899346 Gbit) / (100 Gbit/s) = 0.00858993 s	(Now < 0.01 s = 10 ms)

   Since the allowed down time is 10ms we can use all the bandwidth while the server is down.

   Total time: 16 + 5.12 + 1.6384 + 0.524288 + 0.16777216 + 0.05368709 + 0.00858993 = 23.51273718 s

3. (**20 pts**) Consider flooding in a structured P2P network whose topology is of five dimensional hypercube. Assuming that the node 10001 applies optimal flooding algorithm to send a message m to all nodes in the network, show the message flow.



4. (20 pts) Consider the Chord system, whereby the finger table for a process with process id (pid) p is generated according to the following formula:

$$\mathbf{FT}_{\boldsymbol{p}}[\boldsymbol{j}] = \mathbf{succ}(\boldsymbol{p} + \mathbf{2}^{\boldsymbol{j}-\mathbf{1}} \ \mathbf{mod}\ \mathbf{1024})\ \boldsymbol{j} = \mathbf{1}, \mathbf{2}, \dots, \mathbf{10}$$

a. (10 pts) Fill the following finger table for each process

Process ID	Finger table
83	**128, 128, 128, 128, 128, 128, 203, 266, 339, 722**
128	**203, 203, 203, 203, 203, 203, 203, 266, 415, 722**
203	**266, 266, 266, 266, 266, 266, 339, 339, 466, 722**
266	**339, 339, 339, 339, 339, 339, 339, 415, 722, 806**
339	**374, 374, 374, 374, 374, 374, 415, 492, 722, 878**
374	**415, 415, 415, 415, 415, 415, 454, 722, 722, 895**
415	**454, 454, 454, 454, 454, 454, 492, 722, 722, 938**
454	**466, 466, 466, 466, 492, 492, 722, 722, 722, 998**
466	**492, 492, 492, 492, 492, 722, 722, 722, 722, 998**
492	**722, 722, 722, 722, 722, 722, 722, 722, 750, 83**
722	**745, 745, 745, 745, 745, 806, 806, 878, 992, 266**
745	**750, 750, 750, 806, 806, 806, 878, 878, 83, 266**
750	**806, 806, 806, 806, 806, 806, 878, 895, 83, 266**
806	**878, 878, 878, 878, 878, 878, 878, 938, 83, 339**
878	**895, 895, 895, 895, 895, 938, 958, 83, 128, 374**
895	**938, 938, 938, 938, 938, 938, 998, 83, 128, 415**
938	**958, 958, 958, 958, 958, 998, 83, 83, 203, 454**
958	**998, 998, 998, 998, 998, 998, 83, 83, 203, 454**
998	**1001, 1001, 83, 83, 83, 83, 83, 128, 266, 492**
1001	**83, 83, 83, 83, 83, 83, 83, 128, 266, 492**

b. (5 pts) How does the process with $\boldsymbol{pid} = \mathbf{492}$ reach the data item with key $\boldsymbol{k} = \mathbf{977}$?

$\text{succ}(977) = 998, \quad \text{key} = 977$

$492 \rightarrow 750 \rightarrow 878 \rightarrow 958 \rightarrow 998$

5. (**25 pts**) Consider a decentralized attribute-based naming system, for which we use Hilbert space-filling curves to distribute resources among peers acting as index servers. Resources are files with four (4) normalized attributes; i.e. attributes taking values in [0,1.0]. Attributes are a_1, a_2, a_3 and a_4. A peer with id j acts as an index server for a file k if $j = succ(k)$.

a. (**10 pts**) Calculate the number of indexes supported in this system.

N = 4, k = 4 ⇒ $2^{Nk} = 2^{16} = 65536$

b. (**10 pts**) Consider the query

"$\left(\frac{7}{16} > a_1 \geq \frac{6}{16}\right) \wedge \left(\frac{5}{16} > a_2 \geq \frac{4}{16}\right) \wedge \left(\frac{13}{16} > a_3 \geq \frac{12}{16}\right) \wedge \left(\frac{3}{16} > a_4 \geq \frac{2}{16}\right)$".

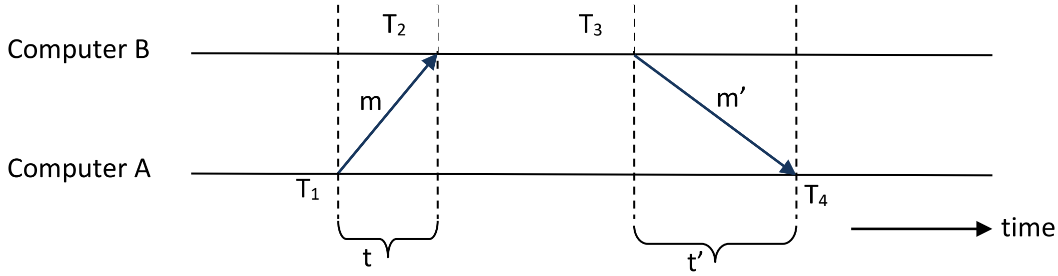
Find the corresponding Hilbert index for the query. You can use the attached python code (hilbert.py) to calculate the index.

15212

c. (**5 pts**) Suppose there are 16 peers and their indexes are 60931, 15992, 34980, 56130, 23519, 42652, 15402, 36858, 53267, 51174, 9876, 35827, 23219, 4098, 41569, 38304. Who will respond to the query in (**b**)?

succ(15212) = 15402

6. (**15 pts**) Two computers (A and B) want to synchronize their clocks using the "symmetric mode" of the NTP.



In order to increase the synchronization accuracy they run the protocol twice and they obtain the following timestamps:

Timestamp	Meaning	First run	Second run
T_1	A sends	13:54:09.950	13:55:09.000
T_2	B receives	13:54:11.750	13:55:10.800
T_3	B sends	13:54:19.000	13:55:15.350
T_4	A receives	13:54:18.200	13:55:14.350

Note the timestamps are given in the format (hour:min:second:millisecond).

$\theta = [(T_2 - T_1) + (T_3 - T_4)]/2$ θ: Offset

$\delta = [(T_4 - T_1) - (T_3 - T_2)]/2$ δ: Delay

a. (**5 pts**) Which clock is faster? Explain your answer.

Clock B is faster since the sending time is greater than the receiving time. ($T_3 > T_4$)

b. (**5 pts**) Which run of the protocol should they use?

	First run	Second run
θ	(1.800 + 0.800) / 2 = 1.3	(1.800 + 1.000) / 2 =1.4
δ	(8.250 – 7.250) / 2 = 0.5	(5.350 – 4.550) / 2 = 0.4

We must choose the smaller valued δ and that is second run.

c. (**5 pts**) Find out the offset between the two clocks.

θ = 1.4s for the second run.