

Part 1a (2 pts)

Each of **6** stations in a network have one packet to transmit. Determine the time that one of the stations is the first to successfully transmit (no collision). At $t=0$ (slot 0), all 6 stations try to transmit and go into the **exponential backoff** part of the transmission protocol (one time slot is 51.2 μ S, therefore, possible transmission times for a station are multiples of 51.2 μ S.)

- **More information on the exponential backoff procedure for this problem is provided on the next page.**
- **Label the stations starting with 0, so your output should contain names of station 0, station 1, station 2, station 3, station 4 and station 5.**

Turn in your answer in table form showing which stations transmit (are scheduled to transmit) during each time slot. Slot 0 is the initial slot (so all 6 stations are listed) for transmissions at $t=0$ (slot 0). Slot 1 is the next slot and it starts at time $t= 51.2 \mu$ s, etc. Lastly, provide the time when the first station is able to successfully transmit and the number of that station (e.g. Station 3 transmits successfully in slot 5 which occurs at time 256 μ S)

Use the input file Project2_part1_rn.txt for the random numbers needed for this problem.

This file contains more than the necessary number of values to successfully complete the problem. If you run out of numbers before finding a solution, then you are making a mistake somewhere. You can solve this problem by hand or use a computer program. If you use a program, provide the source code as an attachment.

Part 1b (2 pts)

For this part each station has 1 packet to transmit. Determine the time when each of the stations make a successful transmission. Need to provide the same information as in part a, except that the information is required for all 6 stations. **Use the input file Project2_part1_rn.txt for the random numbers needed for this problem.**

Part 1c (4pts)

For this part, you have **5** stations in a network and all stations have an endless supply of packets to transmit. After a station successfully transmits a packet, it skips 2 slots (i.e. if it transmits in slot 4, it skips slots 5 and 6 and tries its next packet in slot 7) before attempting to transmit its next packet. A station that successfully transmits a packet will reset its collision count to 0. At what time does the last station successfully transmit its first packet (that is give the minimum time when all stations have successfully transmitted at least one packet). **Use the input file Project2_part1_rn.txt for the random numbers needed for this problem.** If you use a program different from part1A or part1B, submit the source code on Canvas.

Project 2 addendum

Part1a: Determine at what time the first station successfully transmits given 6 stations try to transmit at $t=0$ (in slot 0). Each slot is 51.2 micro seconds, and all transmissions from all stations start at the slot boundary. Use exponential backoff as described in the book. Therefore, after the first transmission attempt, all stations collide. Therefore, they either back off to slot 1 or slot 2. If two or more stations collide in slot 1 (their second collision), they will backoff to slot 2, 3, 4, or 5.

To determine the backoff, the file Project2_part1_rn.txt is available from Canvas. This file contains enough random numbers to determine the time of the first successful transmission (which will be a multiple of 51.2 uS)

For each slot where a collision occurs (two or more stations attempt a transmission), **order the stations from lowest number to highest number**. Then read a number from the input file starting with the lowest station number first. To determine the number of slots to backoff, modulo arithmetic is performed on the number read to determine the number of slots to backoff. The number of collisions that a station has experienced determines the number ($2^{numCollisions}$) to use in the modulo operation.

For example, suppose that a collision occurs in slot #3 and that this collision results in a station having had 2 collisions so far. With 2 collisions, a station will backoff (skip) 0, 1, 2 or 3 slots. So, if the current slot is slot #3, then this station's next transmit time will be in slot #4 (0 for backoff), slot #5 (1 for backoff), slot #6 (2 for backoff) or slot #7 (3 for backoff). The number of slots to back off is determined by reading a number from the input file and performing modulo arithmetic with $2^{numCollisions}$. For the above conditions stated, suppose the number read from the input file for the station is 23. With two collisions, the number of slots to backoff is found from $23\%4$ which equals 3. Therefore, the station will skip the next three slots (slots 4,5 and 6) before attempting a transmission in slot #7.

For your solution, for each time slot write out which stations tried to transmit, and if two or more collide, then provide the next slot they will attempt a transmission. If no station transmits in a slot, then state that fact.