Operational Research / Queuing Systems Set 1

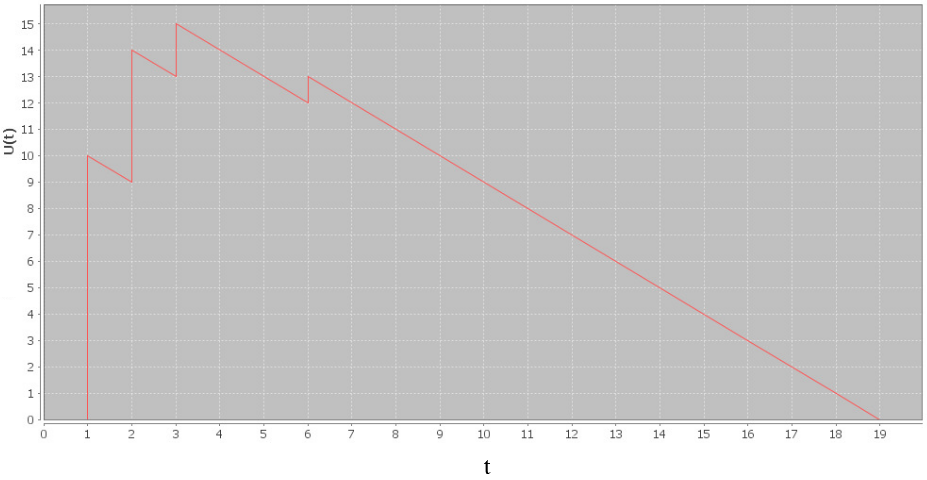
Problem 1

Plot the queuing processes $N(t)$ = *queue length* and $U(t)$ = *unfinished work* for an arrival stream specified by (t_n^+) = (0.5 s, 2 s, 5 s, 6.5 s) and (b_n) =(3 s.u., 2 s.u., 2 s.u., 3 s.u.), and for two cases:

(a) there are S = 2 processors, each serving requests at the speed of v = 1 s.u./s (no grading, processor-bound and time-sharing service mode are assumed),

(b) there is S = 1 processor serving requests at the speed of v = 2 s.u./s (time-sharing service mode is assumed).

Below is an example plot of $U(t)$. What is the value of S? What is the slope of the downward parts of the plot? What would be the slope in case a)? Retrieve the request arrivals stream.



Draw the plots of $U(t)$ for cases a) and b). Explain how they were arrived at. Give an analytic formula.

Explain the no grading, processor-bound and time-sharing service mode assumptions. Discuss the form the plots would take without them.

Discuss the fairness of the comparison of cases a) and b). How can one estimate the offered load factor (ratio of mean demand for service and service supply)?

Name possible comparison criteria of cases a) and b). Is case b) superior? What are the merits of case a)?

Generalize the problem in possible directions. *Can our solution be automated?

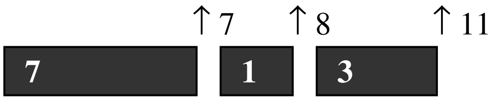
Problem 2

Compare mean system delays in a single-processor queuing system under FIFO and RR with service quantum 2 s (partial use of assigned quantum causes earlier commencement of the next quantum). Processor speed is 1 s.u./s. Three requests, X, Y and Z, of sizes 7 s.u., 1 s.u. and 3 s.u., respectively, arrive simultaneously and queue up in the order (a) XYZ, (b) YZX, (c) XZY.

Explain the workings of both queuing disciplines and premises to apply them in real-world queuing systems.

For RR, what other possibilities exist of handling partial use of assigned quantum?

In analyzing the XYZ scenario for FIFO, the diagram depicted below may be helpful. What diagram obtains for RR?



Analyze in the same way the scenarios b) i c) (perhaps also the other possible), separately for FIFO and RR, and calculate for each the mean system delay of a request.

What general properties of FIFO and RR can be deduced from the comparison of the calculated mean system delays in the respective scenarios? Discuss the value of such an analysis in the context of queuing systems.

Operational Research / Queuing Systems Set 2

Problem 1

A single-processor infinite-buffer queuing system with processor speed v = 1 s.u./s serves a "dense" arrival stream of requests creating a 75% offered load. The total service demand in a one-second observation period is a random variable with standard deviation σ = 0.1 s. What are the chances that the processor can spare half of the second to deal with other (e.g., system) tasks without a backlog of requests forming at the end of the observation period?

Let a random variable $\underline{X}$ represent the total service demand in a one-second observation period. What moments doe the probability distribution of $\underline{X}$ have? What argument yields the shape of this probability distribution?

Recall the relevant probability theorem and the form of $P(a \le \underline{X} \le b)$ that it implies. What values of a and b are of interest in our problem? Give a graphical illustration.

Recall the definition of the Laplace function. What properties of this function are useful when reading its tabulated numerical values (which can be found in textbooks or online)? Look up the necessary values in the tables.

Find the required probability and compare with that of forming a backlog at the end of the observation period.

Problem 2

A queuing system serves on average 800 transactions per second, each transaction on average requiring 5000 elementary operations to complete. An arriving transaction is immediately assigned a processor whose speed is 4,000,000 elementary operations/s. Find the mean number of transactions in system.

How many processors must there be in the system to support the above described operation? Think of a realistic application.

How long on average is the queue of transaction waiting to be processed?

Give the values of the mean request circulation (throughput) and lifetime in the system. What relationship does one now find helpful?

What does the obtained result say about the required number of processors?

Problem 3

A single-processor queuing system with a finite buffer of capacity Q works under offered load $r > 1$. In such a system, the loss fraction L never drops below a certain level – what?

How can one explain on operational grounds the fact that unlimited expansion of the buffer size does not prevent a nonzero loss fraction?

What happens to the processor idle time when Q tends to infinity? How can Little's law be applied to account for the above limit condition?

Operational Research / Queuing Systems Set 3

Problem 1

A queuing system of capacity *Q* handles telemetric reports generated by *J* identical terminals.

Find $J_{\max}$, the maximum number of terminals that can be connected to the processor, and Q_{opt}, optimum buffer capacity (defined as the maximum number of accommodated reports) under the following assumptions:

- each terminal generates on average 20 reports per minute,
- a report contains on average 1800 records of data,
- the system uses a single processor capable of handling 12000 records per second,
- multiple access of the terminals is enabled by a common finite buffer,
- tolerable mean system delay of a report is 1.8 s, and
- tolerable loss fraction due to buffer overflow is 4%.

The table below shows, for various *Q* and offered load *r*, the mean system delay of a report, normalized to the mean report processing time (in boldface), and report loss fraction due to buffer overflow.

Q =	20		21		22		23		24		25	
r =												
0.1	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0
0.2	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0
0.3	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0
0.4	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0
0.5	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0
0.6	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0
0.7	**3.32**	0	**3.32**	0	**3.32**	0	**3.33**	0	**3.33**	0	**3.33**	0
0.8	**4.77**	0	**4.8**	0	**4.84**	0	**4.86**	0	**4.89**	0	**4.91**	0
0.9	**7.23**	0.01	**7.42**	0.01	**7.6**	0.01	**7.76**	0.01	**7.92**	0.01	**8.07**	0.01
1	**10.5**	0.05	**11**	0.05	**11.5**	0.04	**12**	0.04	**12.5**	0.04	**13**	0.04
1.1	**13.5**	0.11	**14.3**	0.1	**15.1**	0.1	**15.9**	0.1	**16.7**	0.1	**17.5**	0.1
1.2	**15.5**	0.17	**16.5**	0.17	**17.4**	0.17	**18.4**	0.17	**19.3**	0.17	**20.3**	0.17
1.3	**16.8**	0.23	**17.8**	0.23	**18.7**	0.23	**19.7**	0.23	**20.7**	0.23	**21.7**	0.23
1.4	**17.5**	0.29	**18.5**	0.29	**19.5**	0.29	**20.5**	0.29	**21.5**	0.29	**22.5**	0.29
1.5	**18**	0.33	**19**	0.33	**20**	0.33	**21**	0.33	**22**	0.33	**23**	0.33

Name the ingredients of a generic system design problem and argue that all of them occur in the formulated problem.

What is the maximum allowable offered load?

What exactly do the boldface numbers in the table represent? How can the tolerable mean system delay be expressed in these terms?

Explain how the last two constraints of the problem influence the feasible values of *Q*.

Suggest some practical steps to raise $J_{\max}$.

Problem 2

Each of 50 terminals connected to a common transceiver generates a request after a think time of average duration $^2/_3$ s. In 80% cases it is a message of average length 1000 bytes, and in 20% cases a control data report of average length 160 bytes. The transceiver works at 1 Mb/s in half-duplex; the average proportion of time it is switched to receive mode is 75% (during that time it is unavailable to the terminals). What is the resulting loss fraction?

Interestingly, we attempt to determine the loss fraction not knowing the buffer size (or even not explicitly assuming its presence) – how can one explain this?

The input data and specified mode of operation translate into the values of a_{m}, b_{m}, and the processor speed and idle time – what are they?

What analytic tool of queuing systems theory permits to link all the above quantities together?

Judge the obtained result and suggest some practical steps to improve it.

Problem 3

In a single-processor queuing system with buffer capacity $Q = 2$ in statistical equilibrium and under offered load $r = 0.75$, we have $p_0 \geq p_1 \geq p_2$. Find the range of possible p_1 values.

Assume that for the considered system, $p_k^+ \equiv p_k$, which in general is not true (on what assumptions is it true?), in this way we are able to link the queue state probabilities with the loss fraction.

What relationship between these probabilities now arises by virtue of the flow conservation equation?

Another relationship is yielded by the normalization constraint, hence we have two relationships (equations) and three unknowns – is it enough to solve our problem?

Instead of the numbers p_k consider the numbers $1 - p_k$, what are the relationships between them?

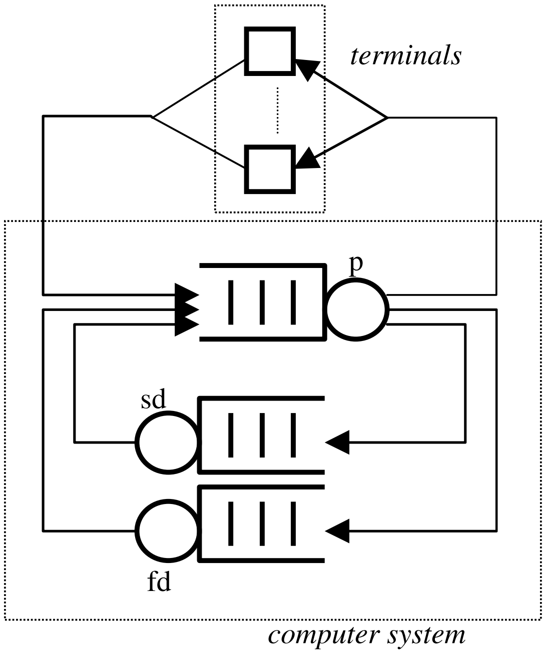
Operational Research / Queuing Systems Practicals Set 4

Problem 1

Each of $J = 30$ computer terminals generates requests that require sequential service at a processor, slow-disk controller and fast-disk controller, as depicted below. On average, a request has to visit these devices $l_p = 21$, $l_{sd} = 12$, $l_{fd} = 8$ times, respectively, whereas average service times there equal $\tau_p = 0.05$ s, $\tau_{sd} = 0.07$ s and $\tau_{fd} = 0.02$ s. Upon notification of service completion for its request, a terminal enters a think time of average $h_m = 15$ s, and subsequently generates another request.

(a) Which device is the bottleneck, and which one is the most overdimensioned? How will his change if the processor is tuned up so that $\tau_p = 0.03$ s?

(b) What processor speedup do we need in order for the mean system delay (request time within the system) to become $d^*_m = 12$ s, and what speedup would ensure $d^*_m = 9$ s?



Draw a typical "lifetime trajectory" of a request within the system.

Are the input data sufficient to determine the mean system delay of a request and why?

Assume a certain request arrival interval at the terminal-system interface. How can it be used to express the offered load of the respective devices? Do we need its numerical value at this point?

What is it that limits the offered loads of the system's devices?

What is the maximum utilization (in %) of the fast-disk controller and why is it so low?

How can one put to good use Little's law in this problem – which part of the illustration is the most convenient to be regarded as a system through which requests circulate?

What is the relationship that the application of Little's law to our system yields between the mean system delay and the rate of request circulation?

Is investment in the processor speed enough to further reduce the mean system delay?

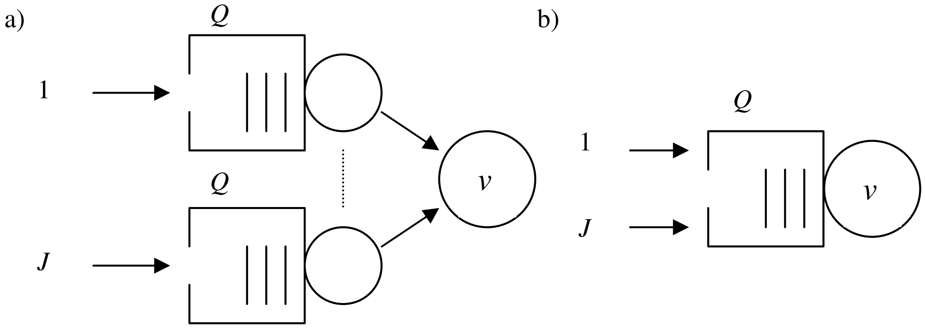
Problem 2

Each user of an M/M/1/Q queuing system generates an arrival stream of documents with mean interval a_{m1} s. Documents have a mean length of b_m bytes. The processor handles documents at v bytes/s. Tolerable are:

- document loss fraction due to buffer overflow not exceeding L_{max}
- mean system delay of a document not exceeding a given multiple c of b_m/v.

Subject to the above, compare the maximum number J_{max} of users and required buffer capacity in two configurations: (a) *dedicated access* with equal-speed virtual processors and separate buffer spaces assigned for the users, and (b) *shared-buffer access* to the processor.

Perform calculations for a_{m1} = 6 s, b_m = 600 bytes, v = 24000 bytes/s, L_{max} = 0.1%, c = 5.



For the queuing systems in cases a) and b) write down expressions for the mean system delay and loss fraction. Compare these expressions.

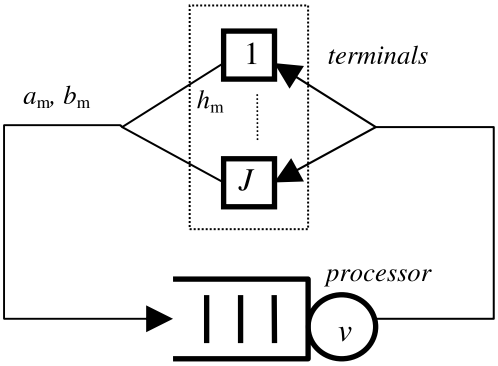
Outline the method of calculation ensuring fulfillment of the two requirements (c and L_{max}), taking Q as an independent variable and noting that a growth of Q makes one requirement less, and the other more stringent.

Perform a numerical calculation of J_{max} and draw appropriate conclusions. What is the reason behind the significant discrepancy between cases a) and b)? What would this discrepancy be for different a_{m1} and b_m?

*For an analogous setting, perform simulations without the M/M/… assumption and compare the results with the previous calculations.

[Problem 3]

A single-processor queuing system interacts with J =10 intelligent terminals in a query-response manner, as depicted below. Having received a response, a terminal generates a new query after a think time of average duration h_m = 4 s. The average number of elementary operations needed to generate a response is b_m = 15000, and processor speed is v = 5000 elementary operations/s. Find the relationship between the proportion of processor idle time and mean *waiting* delay.



Try Little's law again – which part of the illustration is now worth being regarded as a system through which requests circulate?

What is the nature of the required operational characteristic – linear, nonlinear, of what shape? Explain its layout using operational arguments, and suggest a few general conclusions.

Find the flaw in the following argument: the larger the proportion of processor idle time, the smaller offered load and consequently, the smaller mean waiting delay.

Operational Research / Queuing Systems Practicals Set 5

Problem 1

Consider an M/M/*S*/*S* system (also called a loss system since it has no waiting room in buffer) with 150 users. Each user generates on average 10 transactions per second and tolerates *L* ≤ 3% rejections (transactions lost on arrival due to lack of available processors). An average transaction requests the processor to execute 800 elementary operations. The system operator faces a choice between leasing a number of processors of speed 0.5 million elementary operations/s, each at a monthly fee, or four times faster processors leased at a 2.5 times higher monthly fee (only same-type processors can be leased). Which type and how many processors should the operator lease to minimize the total fee while meeting the users' requirements?

For the calculations, use the following table of Erlang-B formula, where *L* values (in %) are given for *S* = 1 to 10 processors, and busy-hour load ranging from $\rho = 0.2$ to 6 erlangs.

$\rho=$ S=	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
1	16.67	28.57	37.5	44.44	50	54.55	58.33	61.54	64.29	66.67	68.75	70.59	72.22	73.68	75
2	1.64	5.41	10.11	15.09	20	24.66	28.99	32.99	36.65	40	43.06	45.86	48.42	50.78	52.94
3	0.11	0.72	1.98	3.87	6.25	8.98	11.92	14.96	18.03	21.05	24	26.84	29.56	32.15	34.61
4	0.01	0.07	0.3	0.77	1.54	2.62	4	5.65	7.5	9.52	11.66	13.87	16.12	18.37	20.61
5	0	0.01	0.04	0.12	0.31	0.63	1.11	1.77	2.63	3.67	4.88	6.24	7.73	9.33	11
6	0	0	0	0.02	0.05	0.12	0.26	0.47	0.78	1.21	1.76	2.44	3.24	4.17	5.21
7	0	0	0	0	0.01	0.02	0.05	0.11	0.2	0.34	0.55	0.83	1.19	1.64	2.19
8	0	0	0	0	0	0	0.01	0.02	0.05	0.09	0.15	0.25	0.39	0.57	0.81
9	0	0	0	0	0	0	0	0	0.01	0.02	0.04	0.07	0.11	0.18	0.27
10	0	0	0	0	0	0	0	0	0	0	0.01	0.02	0.03	0.05	0.08

$\rho=$ S=	3.2	3.4	3.6	3.8	4	4.2	4.4	4.6	4.8	5	5.2	5.4	5.6	5.8	6
1	76.19	77.27	78.26	79.17	80	80.77	81.48	82.14	82.76	83.33	83.87	84.37	84.85	85.29	85.71
2	54.94	56.78	58.48	60.07	61.54	62.91	64.19	65.39	66.51	67.57	68.56	69.49	70.38	71.21	72
3	36.95	39.15	41.24	43.21	45.07	46.83	48.49	50.06	51.55	52.97	54.3	55.57	56.78	57.92	59.02
4	22.81	24.97	27.07	29.1	31.07	32.96	34.78	36.54	38.22	39.83	41.38	42.86	44.29	45.65	46.96
5	12.74	14.51	16.31	18.11	19.91	21.68	23.44	25.16	26.84	28.49	30.09	31.64	33.15	34.62	36.04
6	6.36	7.6	8.91	10.29	11.71	13.18	14.66	16.17	17.68	19.18	20.68	22.16	23.63	25.07	26.49
7	2.83	3.56	4.38	5.29	6.27	7.33	8.44	9.6	10.81	12.05	13.32	14.6	15.9	17.2	18.5
8	1.12	1.49	1.93	2.45	3.04	3.7	4.44	5.23	6.09	7	7.97	8.97	10.01	11.09	12.19
9	0.4	0.56	0.77	1.02	1.33	1.7	2.12	2.6	3.15	3.74	4.4	5.11	5.86	6.67	7.51
10	0.13	0.19	0.28	0.39	0.53	0.71	0.93	1.18	1.49	1.84	2.24	2.68	3.18	3.72	4.31

Why are we not interested here in the mean system delay of a request?

What is the busy-hour load, and when and why is it worth distinguishing from the offered load?

What is the busy-hour load with either type of processors?

State the cost-optimal processor choice. Should the same type be leased if the number of users is doubled?

Problem 2

The arrival stream to an M/M/1 queuing system has mean interarrival interval a_m and mean request size b_m. Processor speed is v s.u./s. Draw a state transition diagram corresponding to the underlying birth-and-death process for the following model specifications:

a) with probability 75% a request whose service has been completed immediately returns to the queue instead of departing from the system,
b) at three or more requests in system, the processor speeds up by 50%,
c) upon termination of a busy period, the processor "goes on vacation" i.e., ignores arriving requests, and only resumes operation when three requests are queued,
d) the processor "goes on vacation" after completion of each request's service; "vacation" duration is exponentially distributed with mean h_m,
e) the processor occasionally breaks down and comes up after a while, whereupon the interrupted service is resumed (all requests arriving during the down-time are queued); the down- and up-times are exponentially distributed with mean f_m i g_m, respectively,
f) requests are admitted in pairs – the first request from a pair is held back until the second one arrives, then the pair is regarded as an arriving request of size equal to the size of the second request.

First formulate some general principles of the construction of state transition diagram based on the operational description, including:

- how to define system state, is the number of requests in system always enough?

- what probability distribution should a transition's time to occurrence have (whose mean value labels the corresponding arrow in the state transition graph)?

- are all the feasible operations within the system reflected in the graph?

*Problem 3

Consider an M/M/1 system with a_m = 10 s, b_m= 10 s.u., and v = 1 s.u./s. On arrival, each request draws its "patience threshold" from exponential distribution with mean $c_m = b_m/v$, and upon its expiration escapes from the queue if still waiting, and if already in service, only departs upon service completion. Find the distribution (p_k) of the number of requests in system, and the fraction L of escaped requests. How to calculate w_m for the non-escaped requests?

What event types determine the mean interval of "death" transitions here? How long is it given there are currently k requests in the system?

Write down the resulting birth-and-death equations. Compare with the M/M/∞ system.

After a routine derivation of p_k based on the birth-and-death equations, determine the escape fraction. Is it the same as the loss fraction considered in the previous problems? Will the flow conservation equation be of any use?